Linux notes

by Uwe F. Mayer <mayer@tux.org>

Introduction

These pages contain notes I wrote for myself during installation of software on my Linux system. I started taking those notes in 1995, and not all the information given here is up-to-date. Also, not all that appears here is written by me, there are quite a few excerpts from README and INSTALL files.

No Warranty

I disclaim all warranties with regard to the information provided herein, including all implied warranties of merchantability and fitness for any particular purpose. In no event shall I be liable for any special, indirect or consequential damages or any damages whatsoever resulting from loss of use, data or profits, whether in an action of contract, negligence or other tortuous action, arising out of or in connection with the use or performance of this information.

1 Acrobat Reader

Files:
*/usr/local/Acrobat?/**
/usr/local/bin/acroread

Sources obtainable at:
`http://www.adobe.com`

Install by running the *INSTALL* script. Version 4 of the reader expects *glibc-2.1* or higher. If you have it, you are done. However, my default glibc2 libraries are version 2.0.7. Hence I edited the binary *acroread* which is in */usr/local/Acrobat4/Reader/intellinux/bin* and change *ld-linux.so.2* to *ld-2.1.2.so.2*, because this is what my glibc-2.1 dynamic loader is called. Then I edited also the wrapper script */usr/local/ Acrobat4/bin/acroread* to make sure the desired libraries are loaded. I added the following lines near the end of the script.

```
# added next three lines to load version glibc-2.1.2 libraries first
LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:/glibc2/2.1.2
```

```
LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:/usr/i686-pc-linux-gnulibc2/lib/2.1.2
export LD_LIBRARY_PATH
```

2 Agrep

Files:
/usr/local/bin/agrep
/usr/local/man/man1/agrep.1gz

Sources obtainable at:
`ftp://ftp.cs.arizona.edu/agrep/`
`http://glimpse.cs.arizona.edu/`

agrep is a new tool for fast text searching allowing errors. *agrep* is similar to *egrep* (or *grep* or *fgrep*), but it is much more general (and usually faster).

Unpack the sources, `cd` into the created directory and run `make`. If desired, edit the Makefile and change the CFLAGS and CC. Install by hand. I made it a slackware package and installed with `pkgtool`.

3 Amaya

Files:
*/usr/local/Amaya/**
*~/.amaya/**
~/.thotrc

Sources obtainable at:
`http://www.w3c.org`

Get the binary distribution for linux and untar from */usr/local*. Link the binaries. Link the HTML-formatted documentation from the index in */usr/local/html*.

4 Apache

Files:
*/usr/local/apache/**

Sources obtainable at:
`http://www.apache.org`

```
CFLAGS="-O2" ./configure --prefix=/usr/local/apache
make
make install
```

Now edit the file */usr/local/apache/conf/httpd.conf* and set the `ServerName` to be 127.0.0.1.

5 Advanced Power Managment

Files:
/usr/X11R6/bin/xapm
/usr/X11R6/man/man1/xapm.1x.gz
/usr/bin/apm
/usr/bin/tailf
/usr/include/apm.h
/usr/lib/libapm.a
/usr/man/man1/apm.1.gz
/usr/man/man1/tailf.1.gz
/usr/man/man8/apmd.8.gz
/usr/sbin/apmd

Sources obtainable at:
`http://sunsite.unc.edu/pub/linux/distributions/slackware/current/source/a`

Edit the *Makefile* and set `CFLAGS="-O2"`, then `make` and `make install` finishes off. Compress the man pages, strip the binaries.

6 AUC TEX

Files:
*/usr/share/emacs/site-lisp/auctex/**
/usr/share/emacs/site-lisp/{tex-site.el,tex-site.elc}
*/usr/info/auctex**

Sources obtainable at:
`http://www.iesd.auc.dk/ amanda/auctex/`
`http://sunsite.auc.dk/auctex/`

AUC TEX is an extensible package that supports writing and formatting TeX files for most variants of GNU Emacs. Unpack the sources, and run

```
make
make lispdir=/usr/share/emacs/site-lisp install
cd doc
gzip -9 auctex auctex-?
cp auctex.gz auctex-?.gz /usr/info
```

Also edit the file */usr/info/dir* to make an entry for AUC TEX.
Change to */usr/share/emacs/site-lisp/* and edit the file *default.el* by adding the lines below.

```
(require 'tex-site)
(setq TeX-auto-save t)
(setq TeX-parse-self t)
(setq-default TeX-master nil)
```

Now byte-compile this file and the file *tex-site.el* with

```
emacs -batch -f byte-compile-batch default.el tex-site.el
```

Finally gzip the files */usr/share/emacs/site-lisp/auctex/*.el*.

7 Aumix

Files:
/bin/aumix
/usr/man/man1/aumix.1.gz
/usr/local/bin/loud

Sources obtainable at:
`http://metalab.unc.edu/pub/Linux/apps/sound/mixers`
`http://jpj.net/ trevor/aumix.html`

I decided to install this without any curses or other library support, so that I just got a small command-line type binary. For this, copy *Makefile.manual* to *Makefile*, and edit it so that you get:

```
OPTS    = -s -Wall
LIBS    =
```

Now make with `make`, and install the binary *src/aumix* and the man page *docs/aumix.1* by hand. Also usefull is the small shells script below, which I called *loud*.

```
#!/bin/sh
aumix -w 100 -c 100
```

8 Autoconf

Files:
*/usr/local/autoconf/**

Sources obtainable at:
`http://sunsite.unc.edu/pub/gnu`

This package is not needed to simply install GNU software, it is needed by the maintainers of GNU software. Hence I uninstalled it.

Autoconf is a tool for producing shell scripts that automatically configure software source code packages to adapt to many kinds of UNIX-like systems. The configuration scripts produced by Autoconf are independent of Autoconf when they are run, so their users do not need to have Autoconf.

```
./configure --prefix=/usr/local/autoconf
make
make install
```

Now compress the info pages, link everything from */usr/local*, and make entries into the info directory file */usr/info/dir*.

9 Automake

Files:
*/usr/local/automake/**

Sources obtainable at:
`http://sunsite.unc.edu/pub/gnu`

This package is not needed to simply install GNU software, it is needed by the maintainers of GNU software. Hence I uninstalled it.

Automake is a tool for automatically generating 'Makefile.in's from files called 'Makefile.am'. Each 'Makefile.am' is basically a series of 'make' macro definitions (with rules being thrown in occasionally). The generated 'Makefile.in's are compliant with the GNU Makefile standards.

The GNU Makefile Standards Document is long, complicated, and subject to change. The goal of Automake is to remove the burden of Makefile maintenance from the back of the individual GNU maintainer (and put it on the back of the Automake maintainer).

```
./configure --prefix=/usr/local/autoconf
make
make install
```

Now compress the info pages, link everything from */usr/local*, and make entries into the info directory file */usr/info/dir*.

10 Backup Procedure

The information given here is for backing up to a 100 MB tape drive, and may be considered obsolete nowadays.

First trim all log files, that is those in */var/adm* and */var/log*. Then empty the *netscape* caches and check */tmp* for old files. No need to back up all this nonsense.

```
insmod ftape.o
mt -f /dev/rft0 rewind
```

Then back up one directory after another. Some directories will be split because they are just too long. If you run this as a shell script insert *sleep 30* between the various commands to allow the ftape to catch up. Note that the directory *dr-xr-xr-x root root /proc* does not need to be backed up. It only contains information about the running processes, and should be created empty after a complete system re-installation. Here is the rough idea (don't do it like this, the files are way too long and won't fit one of my tapes).

```
cd /
tar --exclude usr/lib --exclude usr/local -zcvf /dev/nrft0 usr
tar -zcvf /dev/nrft0 usr/lib
tar -zcvf /dev/nrft0 usr/local
tar -zcvf /dev/nrft0 home
```

```
tar -zcvf /dev/nrft0 bin boot cdrom dev dos etc lib lib-aout lost+found mnt \
                     root shlib sbin tmp var
```

Then rewind the tape and compare with the `-d` flag, for example `tar -zdvf /dev/nrft0` compares the first archive written on the tape. If necessary back up those files that came up with an error and compare again, of course.

11 Bash

Files:
/bin/bash
/bin/sh —> /bin/bash
/usr/bin/bash —> /bin/bash
/usr/bin/bashbug
/bin/static/bash
/bin/static/sh —> /bin/static/bash
/usr/man/man1/{bash.1.gz,bashbug.1.gz,builtin.1.gz}
/usr/man/man3/readline.3.gz
/usr/info/{bash.info.gz,readline.info.gz,history.info.gz}
/usr/doc/bash/

Sources obtainable at:
`ftp://ftp/gnu.ai.mit.edu/pub/gnu/bash/`

The GNU Bourne Again SHell compiles easily, however, it needs a patch or it will break SLIP and other programs that make SUID shell system calls. The patch is in the file *NOTES* that comes with the distribution. Read this file for important update information. Also, with version 2.04 of bash, the file *support/mksignames.c* needs the line `#define _NSIG 64` added.

```
# I keep an older version of bash around, because it is smaller
# cp /bin/bash /bin/bash-1.4.7
patch -p0 < NOTES
CFLAGS="-O6 -fomit-frame-pointer" LDFLAGS="-s" ./configure --prefix=/usr \
     --disable-nls --without-gnu-malloc
make
make check
make install
# under slackware bash should be in /bin, not /usr/bin
install /usr/bin/bash /bin/
ln -sf /bin/bash /usr/bin/
gzip -9 /usr/man/man1/bash.1 /usr/man/man1/bashbug.1 /usr/info/bash.info
# there are two more man pages to install by hand
cd doc
gzip -9 builtin.1 readline.3
install -m 644 builtin.1.gz /usr/man/man1
install -m 644 readline.3.gz /usr/man/man3
# there is one more info page
gzip -9 bashref.info
install -m 644 bashref.info.gz /usr/info
```

There is extra documentation to be had in the file *bash-doc-2.04.tar.gz* from the the same place. I installed parts as follows.

```
tar -zxvf bash-doc-2.04.tar.gz
# now fix links in and compress the HTML-formatted pages
cd doc
for i in *.html; do
  sed -e "s/${i}/${i}.gz/g" $i | gzip -9 > ${i}.gz
  echo $i
done
install -m 644 *.html.gz /usr/doc/bash
```

Make sure you edit */usr/info/dir* and */usr/local/html/index.html*. Finally compile a static version, or leave the old one if it is smaller and you don't need the new features.

```
make distclean
CFLAGS="-O6 -fomit-frame-pointer" LDFLAGS="-s" \
     ./configure --prefix=/usr --disable-nls --without-gnu-malloc \
     --enable-static-link
make
install bash /bin/static
```

12 Binutils

Files:
/usr/bin/{as,ar,c++filt,gasp,gprof,ld,nm,objcopy,objdump,ranlib,size,strings,strip}
/usr/ix86-linux/bin/{ar,as,gasp,ld,nm,ranlib,strip}
/usr/ix86-linux/lib/ldscripts
/usr/ix86-linuxaout/bin/{ar,as,gasp,ld,nm,ranlib,strip}
/usr/ix86-linuxaout/lib/ldscripts
/usr/ix86-coff/bin/ar,as,gasp,ld,nm,ranlib,strip}
/usr/ix86-coff/lib/ldscripts
/usr/man/man1/as.1.gz{ar.1.gz,nm.1.gz,objcopy.1.gz,objdump.1.gz,ranlib.1.gz,size.1.gz strings.1.gz,strip.1.gz,gprof.1.gz,ld.1.gz}
/usr/lib/ldscripts/{elf_i386.x,elf_i386.xbn,elf_i386.xn,elf_i386.xr,elf_i386.xs,elf_i386.xu i386coff.x,i386coff.xbn,i386coff.xn,i386coff.xr,i386coff.xu,i386linux.x,i386linux.xbn,i386linux.xn, i386linux.xr,i386linux.xu}
usr/lib/{libopcodes.so.2.8.1.0.15,libbfd.so.2.8.1.0.15}

Sources obtainable at:
`ftp://tsx-11.mit.edu/pub/linux/packages/GCC/binutils-2.8.1.0.15.bin.tar.gz`
`ftp://sunsite.unc.edu/pub/linux/GCC/binutils-2.8.1.0.15.bin.tar.gz`

Please do back up before you remove things.

```
ln -s i486-linux-gnulibc1 ix86-linux
ln -s i486-linuxaout ix86-linuxaout
rm -f /usr/bin/encaps /usr/bin/nm
tar -zxvf /usr/src/binutils-2.8.1.0.1.5.bin.tar.gz
gzip -9 /usr/man/man1/*.1
```

Now you have the new *gas/binutils* under */usr/bin* and */usr/ix86-linuxaout/bin*. You have to use `/usr/ix86-linuxaout/bin/as` and `/usr/ix86-linuxaout/bin/ld -m i386linux` if you want to use the a.out `as` and `ld` directly.

Further info, not used so far

Lu has changed the target names for Linux/x86 and Linux/x86 (a.out) to *ix86-linux* and *ix86-linuxaout*, respectively. The precompiled binaries are installed under */usr/ix86-linux* and */usr/ix86-linuxaout*. You should make appropriate symbolic links (see above) if you have a different name for the Linux/x86 target, like *i586-unknown-linux*.

If you have an old linux library in the a.out format and you cannot obtain the newer version in the ELF format for whatever reason, you can try "objcopy –remove-leading-char" on the a.out library and see if it can link with your code in ELF. For Sybase client libraries, Lu did

```
# cd sybase/lib
# for f in *.a
do
  objcopy --remove-leading-char $f
done
```

It seems to compile/link fine with libc 5.4.2x. But I don't know if it really works or not. The resulting binaries should work if the libc ABI used by the old library is unchanged in the new libc.

According to Bruce Milner <Bruce.Milner@genetics.utah.edu>:

Just a quick note about your example using objcopy for sybase. The sybase a.out libraries "are" dependent on a changed feature of the libc ABI.

The ctype macros changed to a byte order independent format (little endian). In order to use the sybase libraries, one needs to rename the the ctype table variables in the sybase library files and link in an old version of the libc's "C-ctype.o ctype-info.o" files.

thain@sunquest.sunquest.com put a ELF'ized version of these libraries on

ftp://mudshark.sunquest.com/pub/ctlib-linux-elf

Thanks.

H.J. Lu hjl@gnu.ai.mit.edu 10/31/97

13 Bison

Files:
/usr/bin/{bison,yacc}
/usr/share/{bison.simple,bison.hairy
/usr/info/bison.info.gz
/usr/man/man1/{bison.1.gz,yacc.1.gz}

Sources obtainable at:
`http://ftp.gnu.ai.mit.edu/pub/gnu/bison`

Bison is a parser generator in the style of *yacc*. It should be upwardly compatible with input files designed for *yacc*. Berkeley Yacc is an LALR(1) parser generator. If you don't have *yacc* and

install GNU *bison*, a */usr/bin/yacc* shell script should be created by hand which will allow you to use bison instead.

```
CFLAGS="-O2" ./configure --prefix=/usr --disable-nls
make
make install
strip /usr/bin/bison
gzip -9 /usr/info/bison.info /usr/man/man1/bison.1
#
# If you don't have yacc, do the following
#
echo 'bison -y $*' > /usr/bin/yacc
chmod 755 /usr/bin/yacc
ln -s /usr/man/man1/bison.1.gz /usr/man/man1/yacc.1.gz
```

Make an entry into the info directory file */usr/info/dir*.

14 Brackets

Files:
/usr/local/bin/brackets
/usr/local/src/brackets.c

`Brackets` is a little utility that checks that the number of opening parentheses matches the number of closing parentheses in each paragraph. It distinguishes [, (, and {.

bracket.c

```
/* brackets.c by Uwe F. Mayer <mayer@tux.org> 1994 */

#include <stdio.h>
#include <string.h>
#include <ctype.h>

#define TRUE 1
#define FALSE 0
#define LEFT 123
#define RIGHT 125


int main(argc, argv)
    int argc;
    char *argv[];
{
  int i, old_line_number =1, line_number = 1, parenthesis = 0,
      brackets =0, braces =0, bars =0;
  int empty_line = TRUE;
  char b, c;

  FILE *in;

  void action()
    {
```

```
      if (c != '\n' && c!=EOF)
{
  if (c=='(') parenthesis++;
  if (c==')') parenthesis--;
  if (c=='[') brackets++;
  if (c==']') brackets--;
  if (c==LEFT) braces++;
  if (c==RIGHT) braces--;
  if (c=='|') bars=1-bars;
  if (!isspace(c)) empty_line=FALSE;
}
      else
{
  if (empty_line==TRUE || c==EOF)
    {
      if (c==EOF && empty_line==FALSE) line_number++;
      if (parenthesis !=0 || brackets !=0 || braces !=0 || bars !=0)
{
  if (line_number - old_line_number > 1)
    {
      printf("in lines %d .. %d ",old_line_number,
     line_number-1);
    }
  else
    {
      printf("in line %d ",old_line_number);
    }
  printf("missing ");
  for (i=1; i < 5; i++)
    {
      b=')';
      if (i==2)
{
  parenthesis=brackets;
  brackets=0;
  b=']';
}
      if (i==3)
{
  parenthesis=braces;
  braces=0;
  b=RIGHT;
}
      if (i==4)
{
  parenthesis=bars;
  bars=0;
  b='|';
}
      if (parenthesis!=0)
{
  if (parenthesis < 0)
```

```
    {
      parenthesis=-parenthesis;
      if (i==1) {b='(';}
      if (i==2) {b='[';}
      if (i==3) {b=LEFT;}
    }
  while (parenthesis>0)
    {
      printf("%c",b);
      parenthesis--;
    }
  printf(" ");
}
    }
  printf("\n");
}
      old_line_number=line_number+1;
    }
  empty_line=TRUE;
  line_number++;
}
    }

  if (argc>2)
    {
      printf("usage: brackets [filename] \n");
      return(1);
    }

  if (argc == 1)
    {
     in=stdin;
    }
  else
    {
      if ((in=fopen(argv[1],"r"))==NULL)
{
  printf("Cannot open %s to read\n",argv[1]);
  return(1);
}
    }

  while ((c=getc(in)) != EOF)
    {
      action();
    }
  action();
  printf("Done.\n");

  fclose(in);

  return(0);
```

```
}
```

15 Brakke's Surface Evolver

Files:
*/usr/local/evolver/**

Create the directory */usr/local/evolver* and unpack the sources from there. Fix the ownership of the files, and first compress the documentation.

```
chown -R root.root .
cd doc
for i in *.htm; do
  sed -e 's/\("[^\.]*.htm\)/\1.gz/g' $i > foo && cat foo > $i
done
rm foo
gzip -9 *
```

Then `cd src` and modify the Makefile. Choose `gcc` as the compiler and choose the Linux options for compilation with X-Windows. The three lines should read:

```
CFLAGS= -DLINUX -O3 -fomit-frame-pointer -DOOGL
GRAPH= xgraph.o
GRAPHLIB= -L/usr/X11/lib -lX11
```

Then run `make` to compile. This takes a while. Create the directory */usr/local/evolver/bin* and move the executable *evolver* into it, and remove the *src* directory. Make two entries into the index */usr/local/html/index.html*, one for */usr/local/evolver/doc/default.htm.gz* and another for *manual.ps.gz* in the same directory. Finally edit */etc/profile*. Add the following line.

```
export EVOLVERPATH=/usr/local/evolver/fe:/usr/local/evolver/doc
```

I also downloaded the file *cmd.tar*, gzipped it and put it into the *fe* directory as suggested by Brakke. The file *READ.ME* tells how to make a test run.

It also says in the documentation, that the Evolver expects Geomview (Section) to be at least version 1.6.1 patchlevel 7, or there is an incompatibility.

16 Bzip2

Files:
/bin/static/{bzip2,bzip2recover,bunzip2}
/usr/local/bin/{bzip2,bunzip2,bzip2recover,bzcat}
/usr/local/man/man1/bzip2.1.gz
/usr/local/info/{bzip2.info.gz,bzip2.info-1.gz}

Sources obtainable at:
`http://www.muraroa.demon.co.uk`

```
ftp://ftp.gimp.org/pub/apps/bzip2-0.1pl2.tar.gz
ftp://egcs.cyngus.com/pub/egcs/infrastructure/bzip2-0.9.0c.tar.gz

make
ln -s bzip2 bunzip
ln -s bzip2 bzcat
install -s -m 755 bzip2 bzip2recover bunzip2 bzcat /usr/local/bin/
gzip -9 bzip2.1
install -m 644 bzip2.1.gz /usr/local/man/man1
```

With the latest version comes also documentation in text, TexInfo, and HTML format. I decided to make out of the TexInfo file an info page, however, there are errors. These can easily be avoided by deleting the error causing lines in *manual.texi*, as of this writing these were the lines 41,42, and 100, and by adding a line `@node Top` where you want the info file to start, line 18 seems like a good choice. After that make an entry into */usr/info/dir* and run:

```
makeinfo manual.texi
gzip -9 bzip2.info
install -m 644 bzip2.info.gz /usr/local/info
```

17 Crond and Crontab

Files:
/var/spool/cron/crontabs/root

I have only very few crontab entries. The `crond` daemon is started at boot time and performs commands as outlined in individual crontabs. See the man pages `crontab(1)` and `crontab(5)`. If any action taken by `crond` produces output it is e-mailed automatically to the owner of the crontab. The subject field of the message is that crontab entry that produced the output.

```
# Run the 'atrun' program every 5 minutes
# This runs anything that's due to run from 'at'. See man 'at' or 'atrun'.
0,5,10,15,20,25,30,35,40,45,50,55 * * * *       /usr/lib/atrun
# This informs root about unreasonably long logfiles
0 */2 * * * find /var/adm /var/spool/lp1 -size +100k -exec ls -l {} \;
# This informs root about /tmp files that are older than 3 days
30 */2 * * * find /tmp -atime +3 -exec ls -l {} \;
```

18 CVS

Files:
*/usr/local/cvs/**
${HOME}/.cvspass
/etc/{services, inetd.conf, passwd, group}

Sources obtainable at:
`http://www.cvshome.org`

This is the Concurrent Version System used for software development.

```
CFLAGS="-O2" ./configure --prefix=/usr/local/cvs
make
make install
```

Then compress the man and info pages, strip the binary *cvs*, and link everything from the corresponding places in */usr/local*.
If you wish, you can also run a `make check` after you ran `make`. It should not be run as root, and it takes a long time. In a separate terminal run `tail -f src/check.log` if you want to see the progress.
The files listed above in */etc* need to be modified if one wants to run a cvs server, which I don't. The file */etc/services* needs the line below.

```
cvspserver      2401/tcp
```

The file */etc/innetd.conf* needs the line

```
cvspserver stream tcp nowait root /usr/local/bin/cvs cvs \
   --allow-root=PATH_TO_YOUR_REPOSITORY pserver
```

where I assume that *cvs* on your system is */usr/local/bin/cvs*, and where you need to replace `PATH_TO_YOUR_REPOSITORY` with whatever it is where you have your cvs repository. The files */etc/passwd* (or its shadow) and */etc/group* will need to be modified if you create specific cvs users or groups. You will also need to set up a password file within the cvs tree, together with *reader* and *writer* files, read the documentation.

19 Dejagnu and Expect

Files:
*/usr/local/dejagnu/**

Sources obtainable at:
`ftp://egcs.cygnus.com/pub/egcs/infrastructure/dejagnu-19990614.tar.gz`

```
rm -fr tcl
# get rid of -g flags in the configure script
sed -e 's/\(FLAGS.*=.*\)-g/\1/' configure > foo && cat foo > configure ; rm foo
./configure --prefix=/usr/local/dejagnu --with-tclconfig=/usr/local/lib \
   --with-tclinclude=/usr/local/include --with-tkconfig=/usr/local/lib \
   --with-tkinclude=/usr/local/include --verbose
make
make install
# now prepare and install the info pages
cd dejagnu/doc
makeinfo dejagnu.texi
gzip -9 dejagnu.info*
mkdir /usr/local/dejagnu/info
install -m 644 dejagnu.info*.gz /usr/local/dejagnu/info/
# there is a man page telling people to look for info pages
install -m 644 runtest.1 /usr/local/dejagnu/man/man1
```

```
# the usuall odds and ends
strip /usr/local/dejagnu/bin/*
gzip -9 /usr/local/dejagnu/man/man?/.?
echo "19990614" > /usr/local/dejagnu/VERSION
```

I use the installed *tcl/tk* instead of the one that comes with *dejagnu*. Finally link all the files from the corresponding places in */usr/local*, and make an entry into */usr/info/dir*.

20 Detex

Files:
/usr/local/bin/detex
/usr/local/man/cat1/detex.1.gz

Available at *gopher://arthur.cs.purdue.edu:70/11/Purdue_cs/Users/trinkle/detex.tar*. Unpack the sources into an empty directory, `tar` does not create a subdirectory for the files. Run `make all` to compile it and `nroff -man detex.1l | gzip -9c > detex.1.gz` to create the man page. Then fix the permissions and owner of the files and copy them into the places listed above.

21 Diff

`Diff` is used to compare two files. Giving the `-c` flag asks for context. Here is an example.

hello.world.old

```
#!/bin/sh
echo Hello, World!
echo It's nice to be here.
# This example file needs
# a few lines.
# So I inserted comments.
# That should do.
# But just to make sure
# here are a few more.
```

hello.world

```
#!/bin/sh
echo Hello, World!
echo It's nice to be here.
# This example file needs
# quite a few lines. (See the difference?)
# So I inserted comments.
# That should do.
# But just to make sure
# here are a few more.
```

The output of `diff -c hello.world.old hello.world`

```
*** hello.world.old     Fri Feb  2 23:44:58 1996
--- hello.world Fri Feb  2 23:45:50 1996
```

```
***************
*** 2,8 ****
  echo Hello, World!
  echo It's nice to be here.
  # This example file needs
! # a few lines.
  # So I inserted comments.
  # That should do.
  # But just to make sure
--- 2,8 ----
  echo Hello, World!
  echo It's nice to be here.
  # This example file needs
! # quite a few lines. (See the difference?)
  # So I inserted comments.
  # That should do.
  # But just to make sure
```

22 Ding

Files:
*/usr/local/ding/**
/usr/local/bin/ding
/usr/local/kde/share/applnk/apps/WordProcessing/ding.kdelnk

Sources obtainable at:
`http://www.tu-chemnitz.de/ftp-home/pub/Local/urz/ding/`
`ftp://ftp.tu-chemnitz.de/pub/Local/urz/ding/`

This package prefers to use *agrep* (approximate grep, allows for errors), but will use *egrep* if *agrep* is not installed.
Instructions:

```
cd
mkdir -p /usr/local/ding/bin /usr/local/ding/lib /usr/local/ding/kde
tar -zxvf ding-1.0.tgz
cd ding-1.0
sh install.sh
# when asked specify the two directories created above
# when done, copy the two icons and the kde-link file:
install -m 644 dbook.xpm mini-dbook.xpm ding.kdelnk /usr/local/ding/kde
cd /usr/local/bin/
ln -s /usr/local/ding/bin/ding
# Finally edit and update the file /usr/local/ding/kde/ding.kdelnk to contain
# the correct locations. After you are done:
cd /usr/local/kde/share/applnk/apps/WordProcessing
ln -s /usr/local/ding/kde/ding.kdelnk
```

23 Dip

Files:
*/usr/local/lib/dip/**
/usr/local/bin/slipon
/usr/local/bin/slipoff

`Dip` stands for Dial-up Internet Protocol, and it is a program that allows to connect to the Internet via a serial line. Of course, first one needs to configure a few things, see Section 76.4. `Slipon` and `slipoff` are just two front ends to run `dip` with each one of the script files stored in */usr/local/lib/dip/**. Comments in the script file must start at column one with a #.

slipon

```
#!/bin/sh
E='yes'
if [ $# = 0 ] ; then
    E='no' ; /sbin/dip /usr/local/lib/dip/vandy.dip
  elif [ $# = 1 ] ; then
    case $1 in
      -ppp   ) E='no'; /sbin/dip /usr/local/lib/dip/vandy_ppp.dip ;;
      -p89   ) E='no'; /sbin/dip /usr/local/lib/dip/vandy_89_ppp.dip ;;
      -p91   ) E='no'; /sbin/dip /usr/local/lib/dip/vandy_91_ppp.dip ;;
      -short  | -89   ) E='no'; /sbin/dip /usr/local/lib/dip/vandy_89.dip ;;
      -long  | -91  ) E='no'; /sbin/dip /usr/local/lib/dip/vandy_91.dip ;;
      -utah  ) E='no'; /sbin/dip /usr/local/lib/dip/utah.dip ;;
    esac
fi
if [ $E = 'yes' ] ; then
 echo 'Usage: slipon takes one optional argument to specify the modems to use.'
 echo '  The optional argument specifies modems.'
 echo '  -long or -91  : long-time modem at Vanderbilt at (xxx) xxx-xx91'
 echo '  -short or -89 : short-time modem at Vanderbilt at (xxx) xxx-xx89'
 echo '  -ppp          : all modems at Vanderbilt University using PPP'
 echo '  -p89          : short-time modems at Vanderbilt University using PPP'
 echo '  -p91          : long-time modems at Vanderbilt University using PPP'
 echo '  -utah         : modems at University of Utah'
 echo '  Without the optional argument all modems at Vanderbilt are tried.'
else
    date
    echo
fi
```

slipoff

```
#!/bin/sh
/sbin/dip -k
```

/usr/local/lib/dip/vandy.dip

```
#
```

```
# DIP script to connect to the annex at vanderbilt.edu
# written by Uwe F. Mayer
#
# Set the desired serial port and speed.
  port ttyS1
  speed 115200
# reset ttyS1
# flush out modem responses
# Prepare for dialing and dial.
busy:
  reset
  flush
  print trying long-time modems xxx-xx91
  send ATDTxxxxx91\r
  wait CONNECT 45
  if $errlvl == 0 goto connected
  print BUSY
  reset
  flush
  print trying short-time modems xxx-xx89
  send ATDTxxxxx89\r
  wait CONNECT 45
  if $errlvl == 0 goto connected
  print BUSY
  goto busy
connected:
# do show what's happening
  echo on
  sleep 1
  send \r\r
  wait Starting 40
  if $errlvl != 0 goto error3
# We are connected.  Login to the system.
  sleep 1
  wait Userid: 10
  if $errlvl != 0 goto error4
  sleep 1
  send MYUSERNAME\n
  wait Password? 10
  if $errlvl != 0 goto error5
  sleep 1
  send MYPASSWORD\n
  sleep 1
  send \r\r\r
  wait AccessSwitch 20
  wait > 2
  if $errlvl != 0 goto error6
  sleep 1
  send slip\n
  wait My 10
  wait address 1
  wait is 1
```

```
  if $errlvl != 0 goto error7
  get $remote remote
  if $errlvl != 0 goto errorA
  get $mru remote
  if $errlvl != 0 goto error8
  wait Your 10
  wait address 1
  wait is 1
  if $errlvl != 0 goto error9
  get $locip remote 1
  if $errlvl != 0 goto error10
  get $mtu remote
  if $errlvl != 0 goto errorB
  print
  wait compression 10
  print
  print ************************************************************************
  print Connected to Vanderbilt
  print The Annex's numeric IP address is $rmtip with MRU $mru
  print Your local numeric IP address is $locip with MTU $mtu
  print This connection will be severed automatically after the allotted time
  print ************************************************************************
  print
  default
  mode CSLIP
  goto exit
error1:
  print Modem not responsive
  goto error
error2:
  print Dialing error, possibly just busy
  goto error
error3:
  print Not receiving Annex's welcome message
  goto error
error4:
  print Annex didn't ask for username
  goto error
error5:
  print Annex didn't ask for password
  goto error
error6:
  print Not receiving Annex's command prompt
  goto error
error7:
  print Not receiving the string "Annex address is"
  goto error
error8:
  print Not accepting Annex's address
  goto error
error9:
  print Not receiving the string "Your address is"
```

```
  goto error
error10:
  print Not receiving local IP address
  goto error
errorA:
  print Not receiving MRU
  goto error
errorB:
  print Not receiving MTU
  goto error
error:
  print SLIP connection failed.
  print
exit:
```

/usr/local/lib/dip/vandy_ppp.dip

```
#
# DIP script to connect to the annex at vanderbilt.edu
# written by Uwe F. Mayer
#
# Set the desired serial port and speed.
  port ttyS1
  speed 115200
# reset ttyS1
# flush out modem responses
# Prepare for dialing and dial.
busy:
  reset
  flush
  print trying long-time modems xxx-xx91
  send ATDTxxxxx91\r
  wait CONNECT 45
  if $errlvl == 0 goto connected
  print BUSY
  reset
  flush
  print trying short-time modems xxx-xx89
  send ATDTxxxxx89\r
  wait CONNECT 45
  if $errlvl == 0 goto connected
  print BUSY
  goto busy
connected:
# do show what's happening
  echo on
  sleep 1
  send \r\r
  wait Starting 40
  if $errlvl != 0 goto error3
# We are connected.  Login to the system.
```

```
  sleep 1
  wait Userid: 10
  if $errlvl != 0 goto error4
  sleep 1
  send MYUSERNAME\n
  wait Password? 10
  if $errlvl != 0 goto error5
  sleep 1
  send MYPASSWORD\n
  sleep 1
  send \r\r\r
  wait AccessSwitch 20
  wait > 2
  if $errlvl != 0 goto error6
  sleep 1
  send PPP\n
  print
  print ************************************************************************
  print Connected to Vanderbilt
  print Using Point-to-Point Protocoll
  print This connection will be severed automatically after the allotted time
  print ************************************************************************
  print
  default
  mode PPP
  goto exit
error1:
  print Modem not responsive
  goto error
error2:
  print Dialing error, possibly just busy
  goto error
error3:
  print Not receiving Annex's welcome message
  goto error
error4:
  print Annex didn't ask for username
  goto error
error5:
  print Annex didn't ask for password
  goto error
error6:
  print Not receiving Annex's command prompt
  goto error
error7:
  print Not receiving the string "Annex address is"
  goto error
error8:
  print Not accepting Annex's address
  goto error
error9:
  print Not receiving the string "Your address is"
```

```
  goto error
error10:
  print Not receiving local IP address
  goto error
errorA:
  print Not receiving MRU
  goto error
errorB:
  print Not receiving MTU
  goto error
error:
  print SLIP connection failed.
  print
exit:
```

24 Dvitog3 and Pstog3

Files:
/usr/local/bin/dvitog3
/usr/local/bin/pstog3
/usr/bin/MakeTeXPK
/usr/lib/texmf/mf/base/modes.mf
/usr/lib/texmf/dvips/config.dfaxhigh
/var/texfonts/pk/GThreeFaxHigh/

This section is essentially due to a usenet posting by klaus@snarc.greenie.muc.de and kweidner@physik.tu-muenchen.de

If you use the usual setup, you will probably end up scaling down the 300dpi bitmap fonts available on your system to 204x196dpi needed for faxing. This works, but the quality of the resulting fonts is awful, and Donald Knuth will probably send the International Font Police after you for such cruel abuse of a wonderful typesetting program.

There are three steps you have to take care of to get better results:

- your dvi-to-whatever converter has to use the right font size
- metafont has to be set up to generate fonts of the right resolution when needed.
- both programs have to agree where the fonts are stored.

I use dvips to convert the dvi input to PostScript and then GhostScript to generate g3 fax files. Note that Ghostscript has to be compiled with support for fax devices. The paths used below are those used on my system, you will have to edit them to match your setup.

A shell script takes care of everything from the user's point of view:

```
\$ dvitog3 test.dvi
```

creates the file(s) *test-001.g3* (one per page) in the current directory, which can then be passed on to sendfax. It also creates a file *test.ps* at the fax resolution.

The script looks like this:

dvitog3

```
#!/bin/sh
for i
do
        NAME=‘basename $i .dvi‘
        dvips -P dfaxhigh $NAME.dvi -o $NAME.ps
        gs -sDEVICE=dfaxhigh -sOutputFile=$NAME-%03d.g3 -sNOPAUSE -- $NAME.ps
done
```

This calls dvips with a printer name of `dfaxhigh`. The `-P` flag tells dvips to use the following configuration file:

config.dfaxhigh

```
M GThreeFaxHigh
X 204
Y 196
```

(M defines the configuration Metafont will use to generate the font, X and Y define the resolution in dots/inch.)

You will have to define a Metafont mode for creating the fax fonts, by default it only knows about low-resolution faxes. Put the following definition into */usr/lib/texmf/mf/base/modes.mf*.

Insert for modes.mf

```
[...]
mode_def GThreeFaxHigh =                    % 204 x 196dpi G3fax
   mode_param (pixels_per_inch, 204);
   mode_param (aspect_ratio, 196 / pixels_per_inch);
   mode_param (blacker, 0);
   mode_param (fillin, .2);
   mode_param (o_correction, .2);
   mode_common_setup_;
enddef;
[...]
```

Then create a new metafont base file by running the following commands.

```
inimf "plain; input modes; dump"
mv /usr/lib/texmf/ini/mf.base /usr/lib/texmf/ini/mf.base.bak
mv plain.base /usr/lib/texmf/ini/mf.base
```

Now Metafont knows how to generate fonts at the right resolution. Dvips will check if the needed fonts exist, and if they don't, it will call MakeTeXPK to create them. On my system, the G3 fonts have filenames like '/var/texfonts/pk/GThreeFaxHigh/cmr10.204pk'. The environment variable TEXPKS is used to tell dvips, xdvi and other programs where the fonts are stored, so if they are somewhere else you could set something like the following:

```
TEXPKS=/usr/TeX/lib/tex/fonts/%m/%f.%dpk
```

(%m is replaced with the Metafont 'mode', %f with the font name and %d with the X pixel size.)

If the font doesn't exist, dvips will call a shell script like this: MakeTeXPK <font name> <dpi> <base dpi> <magnification> <mode>.

The MakeTeXPK script was included in my TeX distribution, I have edited it to match my setup, I just added the line for the 204 dpi fonts, and I outcommented the call to `cmmf`, because that seemed not to work.

Changes to MakeTeXPK

```
[...]
   118) MODE=lview;;
   204) MODE=GThreeFaxHigh;;
   300) MODE=CanonCX;;
[...]
#  # Which version of Metafont shall we use?
#  case $NAME in
#    cm*) mf=cmmf;;
#      *) mf=mf;;
#  esac
#  cmmf does not work for some fonts, might be a setup problem, so let us
#  just try mf in any case

  mf=mf

[...]
```

pstog3

This does the same thing as *dvifax*, just for PostScript files.

```
#!/bin/sh
for i
do
  NAME=‘basename $i .ps‘
  gs -sDEVICE=dfaxhigh -sOutputFile=$NAME-%03d.g3 -sNOPAUSE -- $NAME.ps
done
```

25 Egcs

Files:
*/usr/local/gcc/**

Sources obtainable at:
`ftp://egcs.cygnus.com/pub/egcs/releases/`

Gcc and egcs have now merged their efforts, hence the new egcs is now also called gcc. Unpack the sources, and then make a different build directory. Egcs needs the GNU autoconf package.

The testsuite no longer comes with the distribution due to some copyright issues. If desired install the testsuite from a previous distribution. All installation instructions are now in the subdirectory *install*.

```
tar --use-compress-program bzip2 -xvf gcc-2.95.2.tar.bz2
# for the testsuite
ln -s gcc-2.95.2 egcs-1.1.2
tar --use-compress-program bzip2 -xvf egcs-1.1.2.tar.gz \
     egcs-1.1.2/gcc/testsuite
# now continuing with the standard installation
mkdir build
cd build
../gcc-2.95.2/configure --prefix=/usr/local/gcc --disable-nls
make CFLAGS='-O2' LIBCFLAGS='-g -O2' \
     LIBCXXFLAGS='-g -O2 -fno-implicit-templates' bootstrap
# if testing is desired and the testsuite has been installed into gcc/
make check
make install
```

According to the build instructions, setting the flags above in the `make` command reduces disk use by up to 40% during compilation, creates smaller binaries, but still keeps the C-libraries with debugging information.

I prepended */usr/local/gcc/bin* to the PATH variable in */etc/profile*, so gcc of the egcs distribution is now the default gcc compiler. To use the other one, prepend */usr/bin* to your path or use *cc*.

If you have installed the dejagnu package then you can run `make check` for an extensive self test. With egcs-1.1.2 *gcc* unexpectedly fails ten of the self tests, with gcc-2.95.2 it is eleven. *g++* from gcc-2.95.2 has 552 unexpected failures in the egcs-1.1.2 testsuite, but also 43 unexpected passes. The increased number of failures might have to do with the restructuring of the *libg++* library.

Don't forget to compress the info and man pages, to link the info pages from */usr/info/dir*, and to add */usr/local/gcc/man* to the MANPATH. I also want the command *cc* to automatically run *gcc*, hence I added the corresponding link to */usr/local/gcc/bin*.

Running times	AMD 486DX4 133 32 MB RAM	Intel Pentium II 266 32 MB RAM	Intel Pentium II 266 96 MB RAM
version	egcs-1.0.3	egcs-1.1.2	gcc-2.95.2
make bootstrap	2 h 5 min	31 min	31 min
make check	1 h 50 min	29 min	30 min

26 ELF

This is an extract from the ELF–HOWTO.

ELF (Executable and Linking Format) is a binary format originally developed by USL (UNIX System Laboratories) and currently used in Solaris and System V Release 4. Because of its increased flexibility over the older a.out format that Linux currently uses, the GCC and C library developers decided last year to move to using ELF as the Linux standard binary format also.

Background

The aim of this conversion is to leave you with a system which can build and run both a.out and ELF programs, with each type of program being able to find its appropriate breed of shared libraries. This obviously requires a bit more intelligence in the library search routines than the simple 'look in */lib, /usr/lib* and anywhere else that the program was compiled to search' strategy that some other systems can get away with.

The beastie responsible for searching out libraries in linux is */lib/ld.so*. The compiler and linker do not encode absolute library pathnames into the programs they output; instead they put the library name and the absolute path to ld.so in, and leave ld.so to match the library name to the appropriate place at runtime. This has one very important effect—it means that the libraries that a program uses can be moved to other directories without recompiling the program, provided that ld.so is told to search the new directory. This is essential functionality for the directory swapping operation that follows.

The corollary of the above, of course, is that any attempt to delete or move *ld.so* will cause every dynamically linked program on the system to stop working. This is generally regarded as a Bad Thing.

For ELF binaries, an alternate dynamic loader is provided. This is */lib/ld-linux.so.1*, and does exactly the same thing as *ld.so*, but for ELF programs. *ld-linux.so.1* uses the same support files and programs *ldd, ldconfig*, and */etc/ld.so.conf*) as the a.out loader *ld.so* does.

The basic plan, then, is that ELF development things (compilers, include files and libraries) go into */usr/*{*bin,lib,include*} where your a.out ones currently are, and the a.out things will be moved into */usr/i486-linuxaout/*{*bin, lib, include*}. */etc/ld.so.conf* lists all the places on the system where libraries are expected to be found, and ldconfig is intelligent enough to distinguish between ELF and a.out variants. If you have */usr* and */* on different partitions, you'll need to move at least some of the libraries in */lib* to somewhere on the root disk, not on */usr*. Either you can go through the programs that you need to run at system startup or when in single-user mode, and identify the libraries they use, or you can depend on your system/distribution integrator to have done this for you and just move all (err ... some) of the libraries in */lib* to */lib-aout*.

Before you start—Notes and Caveats

You will need to be running a post-1.1.52 kernel with ELF binary format support. You will need:

- *ld.so-1.7.3.tar.gz*—the new dynamic linker
- *libc-5.0.9.bin.tar.gz*—the ELF shared images for the C library and its friends (m (maths), termcap, gdbm, and so on), plus the corresponding static libraries and the include files needed to compile programs with them. *libc 5.2.*something is expected to be released during the lifetime of this HOWTO, and is considerably different from *5.0.9*; if you want to install it, you're on your own, but I'd recommend installing *5.0.9* first and then installing it over the top. There are several parts to *libc.5.0.9* which are not included in *5.2.x* and for which the distribution channels are not entirely set up yet.
- *gcc-2.7.0.bin.tar.gz*—the ELF C compiler. Also includes an a.out C compiler which understands the new directory layout.

- *binutils-2.5.2l.17.bin.tar.gz*—the GNU binary utilities patched for Linux. These are programs such as *gas, ld, strings* and so on, most of which are required to make the C compiler go.

1. Make the new directories that you will move a.out things to

   ```
   mkdir -p /usr/i486-linuxaout/bin
   mkdir -p /usr/i486-linuxaout/include
   mkdir -p /usr/i486-linuxaout/lib
   mkdir /lib-aout
   ```

2. Untar the dynamic linker package *ld.so-1.7.3* in the directory you usually put source code, then read through the *ld.so-1.7.3/instldso.sh* script just unpacked. If you have a really standard system, run it by doing sh instldso.sh, but if you have anything at all unusual then do the install by hand instead.

3. Edit */etc/ld.so.conf* to add the new directory */usr/i486-linuxaout/lib* and */lib-aout*. Then rerun */sbin/ldconfig -v* to check that it is picking up the new directories.

4. Move all the a.out libraries in */usr/*/lib* to */usr/i486-linuxaout/lib*. Note, I said *libraries* not *everything*. That's files matching the specification *lib*.so*, lib*.sa*, or *lib*.a*. Don't start moving */usr/lib/gcc-lib* or anything silly like that around.

5. Now look at */lib*. Leave intact *libc.so**, *libm.so**, and *libdl.so**. If you have symlinks to X libraries (*libX*.so.3**) leave them there too—XView and some other packages may require them. Leave *ld.so**, *ld-linux.so** and any other files starting with *ld*. As for the remaining libraries (if you have any left): if you have */usr* on the root partition, put them in */usr/i486-linuxaout/lib*. If you have */usr* mounted separately, put them in */lib-aout*. Now run *ldconfig -v*.

6. Remove the directory */usr/lib/ldscripts* if it's there, in preparation for installing the binutils (which will recreate it)

7. Remove any copies of *ld* and *as* (except for *ld86* and *as86*) that you can find in */usr/bin*.

8. Some versions of GNU *tar* appear to have problems dealing with symbolic links in the destination directory. Use *cpio* instead of *tar*, it doesn't have this problem.
   `zcat /wherever//libc-5.0.9.tar.gz | cpio -iv` is the magic incantation here, to be executed from the root directory.

9. Install the binutils package. `tar -xvzf binutils-2.5.2.l17.bin.tar.gz -C /` is one perfectly good way to do this.

10. You have now installed everything you need to run ELF executables. Medical experts recommend that VDU workers take regular breaks away from the screen; this would be an opportune moment. Don't forget what you were doing, though; depending on the version of gcc you were previously using, you may have left yourself unable to compile programs in a.out until you unpack the new gcc.

    Backup and remove everything in */usr/lib/gcc-lib/{i486-linux, i486-linuxelf, i486-linuxaout}/* If you use a non-standard gcc driver (eg if you use Gnu ADA), copy that somewhere safe also. Then install the gcc package, again by untarring from root.

11. Some programs (notably various X programs) use */lib/cpp*, which under Linux is generally a link to */usr/lib/gcc-lib/i486-linux/version/cpp*. As the preceding step wiped out whatever version of *cpp* it was pointing to, you'll need to recreate the link:

    ```
    $ cd /lib
    $ ln -s /usr/lib/gcc-lib/i486-linux/2.7.0/cpp .
    ```

    The FSSTND people have once again justified their keep by moving the utmp and wtmp files from */var/adm* to */var/run* and */var/log* respectively. You'll need to add some links dependent on where they currently live, and you may need to make the */var/log* and */var/adm* directories too.

12. If you're intending to continue compiling programs in a.out, this is the appropriate time to install *libc.so.4.7.x*. Untar it from root, as you are now no doubt fully capable of doing without further explanation.

Done! Simple tests that you can try are

```
$ gcc -v
Reading specs from /usr/lib/gcc-lib/i486-linux/2.7.0/specs
gcc version 2.7.0
$ gcc -v -b i486-linuxaout
Reading specs from /usr/lib/gcc-lib/i486-linuxaout/2.7.0/specs
gcc version 2.7.0
$ ld -V
ld version cygnus/linux-2.5.21.14 (with BFD cygnus/linux-2.5.21.11)
  Supported emulations:
   elf_i386
   i386linux
   i386coff
```

Followup on this of course by the traditional "Hello, world" program. Try it with `gcc` and with `gcc -b i486-linuxaout` to check that both the a.out and ELF compilers are set up correctly.

Additional libraries

I created a directory */usr/X11/lib/i486-linuxaout* and moved the aout libraries into there. This also necessitates to fix links in */lib* referring to those moved X-libraries. Then I got the X-libraries from version 3.1.1 in ELF format and put them into */usr/X11/lib*. Then add this directory to */etc/ld.so.conf* and run */sbin/ldconfig -v*.

C++

I also updated **g++** while I was at it. Remove */usr/include/g++* and untar the new distribution from /. Don't forget to run `/sbin/ldconfig -v` again.

27 Emacs

Files:
/usr/bin/{etags,ctags,emacsclient,b2m,rcs-checkin,emacs-20.2,emacs}

*/usr/com/emacs/**
/usr/lib/emacs/
*/usr/share/emacs/20.2/**
/usr/man/man1/{emacs.1.gz,etags.1.gz,ctags.1.gz}
/usr/info/{cl,dired-x,ediff*,emacs*,ccmode*,forms*,gnus,info*,mh-e*,message*,sc*,vip*,*
viper}.gz*

Get the sources from `ftp://prep.ai.mit.edu/pub/gnu` or a mirror site thereof. Unpack the sources, and run `CFLAGS=-O2 ./configure --prefix=/usr`. Save a copy of */usr/info/dir* to keep it from getting overwritten. Then run `make` and `make install`.

Now comes the fixing of the installation. Compress the info pages in */usr/info*, and reinstall the old *dir* file. If you are short on disk space, then remove all **.el* files in sl /usr/share/emacs/20.2/lisp/ that also have a compiled version (**.elc*). Then compress the man pages, and finally remove */usr/share/emacs/20.2/site-lisp/* and replace it by a link to */usr/share/emacs/site-lisp/*. Finally make sure all permissions and ownerships are correctly set (run the commands `chmod -R a+rX` and `chown -R root.root` on the appropriate directories).

28 Fast-Find

Files:
/usr/local/bin/ff
/usr/local/sbin/makeffdata
/usr/local/lib/ff.data.gz

The file *ff.data.gz* contains a complete listing of the root directory tree created by a run of `makeffdata`, and `ff filename` searches this file. This is much faster than running `find / -name filename`. The same thing can be achieved by the GNU *locate* and *updatedb* utilities.

The file *makeffdata*:

```
#!/bin/sh
cd /
find . -exec ls -ld {} \; | gzip -9c > /usr/local/lib/ff.data.gz
```

The file *ff*.

```
#!/bin/sh
if [ "X$1X" = "X-lX" ];
then
 if [ $# != 2 ];
  then
   echo "usage: ff [-l] regexp"
  else
   gzip -dc /usr/local/lib/ff.data.gz | grep $2 | sort -u | less
 fi
else
 if [ $# != 1 ];
  then
   echo "usage: ff [-l] argument"
```

```
  else
   gzip -dc /usr/local/lib/ff.data.gz | grep $1 | sed -e 's|^[^/]*||' | \
   sort -u| less
 fi
fi
```

29 Fdisk

`Fdisk` is a program used to partition a hard drive. There is a DOS version and a Linux version. Use each version only to make partitions for the respective operating system. It is necessary to boot off a floppy to use this program. Notice that DOS allows only one primary and one extended partition. The extended partition can be split into many logical drives. If there is more than one hard drive, then set one drive on each IDE channel to be the master, and set the other to be the slave. The IDE CD-ROM must be the last drive. Use the fastest drive as your master drive. Finally, remember that `lilo` can only boot from cylinders with a number less than 1024. My new Fujitsu has 5086 cylinders, even as there are only 635 reported, they get grouped into groups of eight, so I have to have a boot cylinder less than 127.
The information below is old!

I created one primary partition for DOS. My hard drive happens to have bad blocks so I skipped the next four cylinders. Chance has it that this is were the bad blocks are. Use `SCANDISK` in DOS to check your disk. Warning: I think this program eats my drive. `SCANDISK` only scans formatted drives. Use `FORMAT /S C:` to make C a bootable partition (the `/S` switch copies the DOS kernel).

For the Linux partitions also boot off a floppy. The hard drive should not be mounted when you use `fdisk`. First I set up the swap partition. Run `fdisk`. My sequence of keystrokes were:
`p` to print the partition table;
`n` to create a new partition;
`p` to make it a primary partition;
`3` for partition number 3;
`265` for the first cylinder to use (my DOS ends at 260 and I left four empty for the bad blocks I have there);
`+40M` for 40 MB;
`t` to set the type;
`82` for Linux swap;
`p` to print the resulting partition table; and finally
`w` to write and exit fdisk.

Creating the root partition is very similar. Use the rest of the disk. There is no need to change the type as it automatically assigns the correct one. Then you need to create the file systems on the new partitions (like formatting under DOS). For the swap partition the command is `mkswap -c /dev/hda3 41328`. The `-c` flag stands for checking the disk for bad blocks, and the number `41328` is the size of the partition in blocks. You get the number from the output of `fdisk`. For the main partition I use the ext2 file system. The command is `mke2fs -c /dev/hda4`. This automatically makes a directory */lost+found* and reserves 5% of the disk for the superuser (root). Partitions

in this format can be checked with the `badblocks -v -w device blocksize` command. Be very careful, **this erases all data on the disk!** Finally, update */etc/fstab*.

30 Fetchmail

Files:
/usr/local/bin/fetchmail
/usr/local/man/man1/fetchmail.1.gz
/usr/local/doc/fetchmail/{fetchmail.FAQ.html.gz, sample.rcfile.gz}

Obtainable from any Linux site in *system/mail/pop*. Compiles with `./configure` and `make CFLAGS="-O2 -m486"`. Then strip the binary, do a `make install`, compress *fetchmail.FAQ.html* and *sample.rcfile* and put them into the place listed above. Link the FAQ from the index in */usr/local/html*. Finally, enable *sendmail* in the */etc/rc.d/rc.M* file or fetchmail will loose all your mail!

Remove the old installation of *popclient* which *fetchmail* replaces. *Popclient* lost some of my mail as it transferred it but never but it into the local inbox!

31 Findlinks

Files:
/usr/local/sbin/findlinks

This self-documenting shell script is useful to find links in subdirectories pointing to a specific other directory.

```
#!/bin/sh
if [ "$2" = "" ]; then
  echo
  echo "Usage:       findlinks name dir1 [dir2] [dir3] [...]"
  echo
  echo "Description: This shell script will find any links to name in dir1, d
ir2, ..."
  echo
  echo "Example:     To find all links in /usr/local/bin and /usr/local/lib"
  echo "             pointing to /usr/local/automake run the command:"
  echo "             findlinks /usr/local/automake /usr/local/bin /usr/local/
lib"
  echo
  exit 1
fi
name=$1
shift
find $* -type l -exec ls -l {} \; | \
  fgrep ${name} |\
  sed -e 's|-> '${name}'.*||' |\
  cut -b 56-
```

32 Freeamp

Files:
/usr/local/bin/freeamp
*/usr/local/lib/freeamp/**
*/usr/doc/freeamp-1.3.1/**
/var/adm/packages/freeamp

Sources obtainable at:
`http://www.freeamp.org`

I downloaded the binary distribution *freeamp-1.3.1-1.rh5x.i386.rpm*. This is a *glibc2*-based distribution. I made it into a Slackware package and installed with *pkgtool*, see Section 102.1. Put the distribution into an empty directory and run the commands below.

```
rpm2cpio freeamp-1.3.1-1.rh5x.i386.rpm | cpio -ivmd
chmod -R a+rX usr
# see explanation below for the next few commands
rm usr/local/lib/freeamp/plugins/alsa.pmo
rm usr/local/lib/freeamp/plugins/esound.pmo
rm usr/local/lib/freeamp/plugins/ncurses.ui
strip usr/local/lib/freeamp/plugins/* usr/local/bin/freeamp
tar -zcvf freeamp.tgz usr
echo "CONTENTS: freeamp" >  disk___
echo "freeamp:  FreeAmp version 1.3.1" >> disk___
echo "freeamp:" >> disk___
pkgtool
```

Now choose to install from the current directory. Upon running freeamp the first time, I got the following messages. Hence I removed the offending files, as stated in the installation instructions above. Also, none of the binaries are stripped. This would make it a 5 MB installation, versus a final 1.4 MB installation.

```
libasound.so.0: cannot open shared object file: No such file or directory

This is *not* a fatal error. The plugin 'alsa.pmo' in FreeAmp's plugin
directory could not be loaded. FreeAmp will use the default PMO.
To get rid of this warning either fix the problem, or delete the
'alsa.pmo' file from FreeAmp's plugin directory.
libesd.so.0: cannot open shared object file: No such file or directory

This is *not* a fatal error. The plugin 'esound.pmo' in FreeAmp's plugin
directory could not be loaded. FreeAmp will use the default PMO.
To get rid of this warning either fix the problem, or delete the
'esound.pmo' file from FreeAmp's plugin directory.
/usr/local/lib/freeamp/plugins/ncurses.ui: undefined symbol: wcolor_set
```

33 Freetype

Files:
*/usr/local/freetype1/**

Sources obtainable at:
`http://freetype.sourceforge.net`

This is a free font rendering library and also comes with a utility to transform True-Type-Fonts to bitmap fonts. I got *freefont-1.3.1.tar.gz* and *freefont-contrib.tar.gz*.

```
tar -zxvf freefont-1.3.1.tar.gz
tar -zxvf freefont-contrib.tar.gz
mv freefont-contrib freefont/contrib
CFLAGS="-O2" ./configure --disable-nls --prefix=/usr/local/freetype1
make
make install
cd contrib/ttf2bdf
CFLAGS="-O2" ./configure --disable-nls --prefix=/usr/local/freetype1
make
make install
cd ../ttf2pfb
CFLAGS="-O2" ./configure --disable-nls --prefix=/usr/local/freetype1
make
make install
strip /usr/local/freetype1/bin/*
gzip -9 /usr/local/freetype1/man/man1/*
```

Now link all from the right places in */usr/local/*.

Usage of ttf2bdf

I specifically installed this package for the use of this program, to get fonts that work well with Netscape, see Section 76.4. I was missing arial fonts, and helvetica, times, and courier at certain sizes. For example, to get the arial font size 11, which I will also use as a substitute for helvetica size 11, I use the Microsoft file *arial.ttf*.

```
ttf2bdf -r 75 -p 11 -o arial11.bdf -w medium -t arial -s r -k normal arial.ttf
mkfontdir
```

That *mkfontdir* command created *font.dir*, which in absence of other fonts in this directory reads as:

```
1
arial11.bdf -freetype-arial-medium-r-normal--11-110-75-75-p-55-iso10646-1
```

Then I wrote a file *fonts.alias*. It contains:

```
-adobe-arial-bold-r-normal--11-110-75-75-p-59-iso8859-1 \
        -freetype-arial-bold-r-normal--11-110-75-75-p-59-iso10646-1
-adobe-helvetica-bold-r-normal--11-110-75-75-p-59-iso8859-1 \
        -freetype-arial-bold-r-normal--11-110-75-75-p-59-iso10646-1
```

The lines ending on a backslash need to be connected, they just did not fit here on the same line. I put all the files into the directory */usr/local/lib/X11/fonts/75dpi* and added that directory to the X11 font path in */var/X11R6/lib/XF86Config*. Notice that changing the encoding information is really a hack, but it works for Netscape.

34 Fstab

Files:
/etc/fstab

The file */etc/fstab* contains information about mount points for the file systems. For a file system to be mounted at boot time it has to be listed here. It also allows to specify properties other than the defaults. Finally, all systems listed here can be mounted by just typing `mount` followed by the mount point. The lines contain device, mount point, type, options, and two numbers. The first number is used for backup purposes, it gives the number of days between backups, should be zero for swap partitions. I do not use any programs that use that number. The second number is used by `fsck` as a pass number. The program `fsck` checks existing file systems. If there is more than one file system, then the pass number can be used to specify the order in which they are checked. By default file systems are checked in parallel. The root system must have pass number 1, and two file systems on the same drive should have different numbers. The options are as follows. **rw** for read/write, **ro** for read only, **suid** to allow set-user or set-group bits to be effective, **dev** to interpret special character or block devices, **exec** to allow for executable binaries, **auto** to be able to mount it with `mount -a`, **user** to allow ordinary users to mount it, **async** to provide buffering of read/write, and **defaults** stands for **rw, suid, dev, exec, auto, nouser,** and **async**. Notice that DOS partitions have no SUID or SGID bits.

```
/dev/hda2    /             ext2      defaults                              1 1
/dev/hda5    /usr          ext2      defaults                              1 2
/dev/hda6    /home         ext2      defaults                              1 2
none         /dev/pts      devpts    gid=5,mode=620                        0 0
none         /proc         proc      defaults                              0 0
/dev/hda3    swap          swap      defaults                              0 0
/dev/hdc     /cdrom        iso9660   ro,suid,dev,exec,auto,user,async      1 0
/dev/fd0     /a/linux      ext2      rw,suid,dev,exec,noauto,user,async    1 0
/dev/fd0     /a/minix      minix     rw,suid,dev,exec,noauto,user,async    1 0
/dev/fd0     /a/dos        msdos     rw,nosuid,dev,exec,noauto,user,async  1 0
/dev/fd0     /a/win        vfat      rw,nosuid,dev,exec,noauto,user,async  1 0
/dev/hda1    /c            vfat      rw,nosuid,dev,exec,auto,user,async    1 2
/dev/sda1    /zip/linux1   ext2      rw,suid,dev,exec,noauto,user,async    1 0
/dev/sda1    /zip/dos1     msdos     rw,nosuid,dev,exec,noauto,user,async  1 0
/dev/sda1    /zip/win1     vfat      rw,nosuid,dev,exec,noauto,user,async  1 0
/dev/sda4    /zip/linux4   ext2      rw,suid,dev,exec,noauto,user,async    1 0
/dev/sda4    /zip/dos4     msdos     rw,nosuid,dev,exec,noauto,user,async  1 0
/dev/sda4    /zip/win4     vfat      rw,nosuid,dev,exec,noauto,user,async  1 0
```

35 Ftape

Ftape is part of the kernel since Linux-2.0.0.

Excerpt of the Install-guide

As said before, the *ftape.o* module will only run with the kernel version it's compiled on, if you get an error message saying that the kernel version doesn't match the current kernel, remove the file kernel-version.h (or do a `make clean`) and run `make` again. There should be **no** warnings. **All** compiler warnings / errors should be investigated and reported. [I get one warning which I ignored so far.]

You'll need to log in as root to load the driver with `insmod /boot/ftape.o`. The messages that ftape generates are caught by the syslog daemon and can be found in */var/adm/messages*. If everything went well you got a couple of lines including a copyright notice and some timing information.

You have to reload the driver *every* time you boot the system. (Just as with the modules utilities, it's a good idea to have *ftape.o* on the root partition. The filesystem standard suggests somewhere under /boot.)

Now put a tape in the drive and wait for the drive to stop making noise. (Most tape drives start calibrating the head when a new tape is loaded.) Now try if the drive is recognized: `mt -f /dev/rft0 rewind`. This will try to open the device. No tape motion will follow if the tape is already rewound. Because initially *tracing* is set to some value greater than zero, some debugging output (printk) is generated on the console or in */var/adm/messages*. The messages generated when opening the device can start with some "timeout on Acknowledge" lines. This depends on the trace-level, and is normal as long as the driver is probing for a tape drive. If they keep appearing after the device is opened something is wrong.

If all goes well, messages will appear saying something about "drive-wakeup method: xxxx" and "tape drive type: yyyy". The next test verifies if the drive does respond to some simple commands: `mt -f /dev/ftape reten`. The tape should wind completely to the end once and back to the beginning again. Depending on the tape used this may take a couple of minutes. If there are no error messages in the log everything is going fine. Now we are going to try to read some data from the tape. [Make sure a formatted tape is in the drive !] `mt -f /dev/ftape fsf 0`. This causes the driver to read the header segment and display some information on the tape in the drive (look in the kernel logfile). `Mt` may fail with an I/O error if the tape contains no file marks or if the operation failed otherwise. Look in the log messages for errors. This may indicate that you're using the wrong dma channel with an FC-10 controller or other special configuration.

Now it's time to try the write operation. Put a formatted tape in the drive (write protect removed) and initialize the tape for usage with the ftape driver: `mt -f /dev/ftape erase`. This command writes a Linux tape label so you can recognize the tape when using other operating systems, and it clears all file marks on the tape (No real erasure takes place, all data is still accessible). This command must be issued before file marks can be written to tape !

At this point you're ready to use the driver to try a simple backup of the kernel sources. We'll be using gnu tar to write and verify the data: `tar clvf /dev/ftape /usr/src/linux/*`. You'll see the file names coming by as the data is written to tape. If the operation succeeds without errors a verification can be done: `( cd / ; tar dvf /dev/ftape )`. The data on tape will be compared with the files on disk and any mismatches will be reported.

At this point the driver seems to be cooperating with the tape drive and you'll have to determine the method you'll use to make the backups. Unix (and therefor Linux) has a lot of tools to do this and it is a matter of taste which you want to use. Some of the options are: `cpio, tar, dd` and

`dump/restore`. One final warning: If you are using compression when writing data to tape (e.g. tar with the 'z' option) you may not be able to recover anything if the tape gets damaged !

A short introduction to file marks

File marks are written to tape to separate individual files (Note that one 'tar' file may contain many data files!). So several files can be written to tape and individually retrieved. The way `ftape` implements these file marks allows almost random access to individual files (instead of sequentially reading the entire tape until ones finds the right file mark). Every time the device is closed after writing some data two file marks are written to tape. If the non-rewinding device is used the tape is rewound so it is positioned right between these two marks. If another file is written to tape the second marks is overwritten but the first stays causing succeeding files to be separated by exactly one file mark. The following should make this clear: After writing one file to the non-rewinding tape device the tape layout will be:
`<file-1> <eof> <eof>`
When another file is written the layout will be:
`<file 1> <eof> <file 2> <eof> <eof>`
If we want to add another file to this tape after it has been rewound, there are two options: First to skip 2 file marks to position right between the last two marks. In this case we have to know how many files to skip. The other solution is to seek for the double file mark and position in between. This is done with `mt eom`. Remember to use the **non**-rewinding device when positioning the tape or it will be rewound when closing !

Appending a file to a tape:

```
mt -f /dev/nftape eom
tar cvf /dev/ftape <files>
```

Reading the first and third files from tape:

```
mt -f /dev/ftape rewind
tar xvf /dev/nftape
mt -f /dev/nftape fsf 2
tar xvf /dev/nftape
```

The best way to understand what is happening is to picture the tape layout and count the file marks that have to be skipped.

Warning: Keep in mind that file marks aren't written to tape until the header segment is updated. For non-rewinding devices this will not happen automatically: Make sure that either the last write is to a rewinding device or issue an `mt rewind` or `mt rewoffl` to flush the buffers. The last command will put the drive offline until you reload the cartridge (or the driver). While the drive is offline all commands will fail.

36 GAP

Files:
*/usr/local/gap4beta/**
/usr/local/bin/{gap, gac}

Get the sources from `http://www-groups.dcs.st-and.ac.uk/ gap/`, which is at the St. Andrews University in Scottland, or get them from `http://wwww.math.rwth-aachen.de/LDFM/GAP/`. Create directory */usr/local/gap4beta* and unpack the sources from there, the fix-files must be unpacked last, in the right order. Then cd into *src* and run `make i386-ibm-linux-gcc2`. After this cd into the *../bin/i386-ibm-linux-gcc2* directory and delete all the *.o files, and strip the executable *gap*. Then `cd ..` and edit the script file gap.sh, and finally place links to it and to the gap-compiler *i386-ibm-linux-gcc2/gac* into */usr/local/bin*.

37 Geomview by The Geometry Center

Files:
*/usr/local/Geomview/**

I tried to compile the source distribution, but no luck. So I installed the binary dostribution.

Unpack the sources from */usr/local* and afterwards run `chown -R root.root Geomview`. As this is a binary distribution, no compiling is necessary. `Cd` into *Geomview/man* and `rm -rf cat?`. There is no need to have both formatted and unformatted man pages. Run `cd Geomview; ./installbin`. This will ask a few questions, where Geomview is installed, where to put the binaries, where to put the man pages, if you have Mathematica or Maple. Hit `Enter` to accept the default answers for all those questions. Finally, do not send a message (we are not on the Internet). Then run `./installbin -clean`. I moved all the man pages and binaries back into the Geommview file tree.

The man pages were not compressed, so go to */usr/local/Geomview/man/man1,3,5* and `gzip` them.

`Cd /usr/local/Geomview/doc` and rename *geomview* to *geomview.info*, similarly rename the files *geomview-?* where ?=1,...,5. Then compress the info files, the ps and the tex files with `gzip -9`. Finally move the info files to */usr/local/Geomview/info* and make an entry into */usr/info/dir*. The HTML formatted documentation needs to be changed so that the links inside are rewritten to point to .html.gz files instead of .html files, use `sed` to do this. After that make two entries into the general index */usr/local/html/index.html*, one for *Geomview/html/geomview_toc.html.gz* and another for *Geomview/doc*.

38 Gettext

Files:
*/usr/local/gettext/**

Sources obtainable at:
`ftp://alpha.gnu.mit.edu/pub/gnu`

This package is not needed to simply install most GNU software, it is needed by the maintainers of GNU software. Hence I uninstalled it. However, glibc2 (see Section 43.1) does use it, so you need it if you want internationalization of your C library.

Usually, programs are written and documented in English, and use English at execution time to interact with users. This is true not only of GNU software, but also of a great deal of commercial and free software. Using a common language is quite handy for communication between developers, maintainers and users from all countries. On the other hand, most people are less comfortable with

English than with their own native language, and would prefer to use their mother tongue for day to day's work, as far as possible. Many would simply *love* to see their computer screen showing a lot less of English, and far more of their own language.

GNU 'gettext' is an important step for the Translation Project, as it is an asset on which we may build many other steps. This package offers to programmers, translators and even users, a well integrated set of tools and documentation. Specifically, the GNU 'gettext' utilities are a set of tools that provides a framework within which other free packages may produce multi-lingual messages. These tools include a set of conventions about how programs should be written to support message catalogs, a directory and file naming organization for the message catalogs themselves, a runtime library supporting the retrieval of translated messages, and a few stand-alone programs to massage in various ways the sets of translatable strings, or already translated strings. A special mode for GNU Emacs also helps ease interested parties into preparing these sets, or bringing them up to date.

```
CFLAGS="-O2" ./configure --prefix=/usr/local/gettext
make
make install
```

Now compress the info pages, link everything from */usr/local*, and make entries into the info directory file */usr/info/dir*.

39 Giftrans

Files:
/usr/local/bin/giftrans
/usr/local/man/cat1/giftrans.1.gz

`Giftrans` transforms graphics in gif format from gif87 to gif89, making one color transparent. Useful for world wide web pages. Unpack the sources and compile via `gcc -O2 -Wall giftrans.c -o giftrans`. Change the permissions and the ownership to what it should be and install the binary into */usr/local/bin* and the unformatted man page into */usr/local/man/man1*. Use `man giftrans` to format the man page and then remove the unformatted version.

40 Gimp

Files:
/usr/local/doc/Gimp-Manual-1.0.0.pdf

Gimp is part of the official Slackware release 7.0, so the files are not listed above. What is listed above is the additional manual I installed.

Old Gimp information

Files:
*/usr/local/glib/**
*/usr/local/gtk+/**
*/usr/local/gimp/**
/etc/ld.so.conf

Sources obtainable at:
`http://www.gimp.org:`

```
            glib-1.2.1.tar.gz
            gtk+-1.2.1.tar.gz
            gimp-data-extras-1.0.0.tar.bz2
            gimp-1.0.4.tar.bz2
            GimpUserManual-1.0.0.pdf
```

40.1 The support library glib

While version 1.2.2 is the latest stable release as of this writing, it did not compile on my system, hence I am running version 1.2.1.

```
tar -zxvf glib-1.2.1.tar.gz
mkdir build
cd build
CFLAGS="-O2" ../glib-1.2.1/configure --prefix=/usr/local/glib\
  --disable-threads
make
make install
```

Now compress the info files and the man page, and make an entry into */usr/info/dir*. Link everything from */usr/local*. Add the line `/usr/local/glib/lib` to the file */etc/ld.so.conf* and run `ldconfig -v`.

40.2 The support library gtk+

As I have only glib version 1.2.1, I am also running gtk+ at version 1.2.1.

```
tar -zxvf gtk+-1.2.1.tar.gz
mkdir build
cd build
CFLAGS="-O2" ../glib-1.2.1/configure --prefix=/usr/local/gtk+
make
make install
```

Now compress the info files and the man page, and make an entry into */usr/info/dir*. link everything from */usr/local*. Add the line `/usr/local/gtk+/lib` to the file */etc/ld.so.conf* and run `ldconfig -v`.

40.3 Gimp

I chose to install Gimp, the extra patterns, and the user manual.

```
bunzip2 < gimp-1.0.4.tar.bz2 | tar -xvf -
mkdir build
cd build
CFLAGS="-O3" ../gimp-1.0.4/configure --prefix=/usr/local/gimp
make
```

```
make install
mkdir /usr/local/gimp/doc
cp GimpUserManual-1.0.pdf /usr/local/gimp/doc
```

In the same fashion install the extras, if desired.

```
bunzip2 < gimp-data-extras-1.0.0.tar.bz2 | tar -xvf -
mkdir build
cd build
../gimp-data-extras-1.0.0/configure --prefix=/usr/local/gimp
make
make install
```

Now compress the man pages. Link everything from */usr/local*. Add `/usr/local/gimp/lib` to the file */etc/ld.so.conf* and run `ldconfig -v`.

41 Ghostscript

Files:
*/usr/local/ghostscript/**

Default installation has these:
Files:
/usr/local/bin/{bdftops, dvipdf, font2c, gs, gsbj, gsdj, gsdj500, gslj, gslp, gsnd, pdf2dsc, pdf2ps, printafm, ps2ascii, ps2epsi, ps2pdf, ps2ps, wftopfa}
*/usr/local/share/ghostscript/** (includes documents, examples, postscript files)
/usr/local/man/cat1/{gs.1.gz, ps2epsi.1.gz, pdf2dsc.1.gz, ps2pdf.1.gz, pdf2ps.1.gz, ps2ps.1.gz, ps2ascii.1.gz}

The sources are available from `http://www.cs.wisc.edu/~ghost/index.html`, which is the home page of the Aladdin Ghostscript release. Currently you need the following files:

```
ghostscript-6.50.tar.gz
ghostscript-6.50jpeg.tar.gz
ghostscript-6.50libpng.tar.gz
ghostscript-6.50zlib.tar.gz
ghostscript-fonts-other-6.0.tar.gz
ghostscript-fonts-std-6.0.tar.gz
```

Remove your old installation first using `pkgtool` and don't forget the empty directories, or remove all the files directly. Note: I install everything into */usr/local/ghostscript/*. Before compiling the sources the fonts need to be put into the directory */usr/local/ghostscript/share/ghostscript*. Create this directory if it's not there and `cd` into it. Unpack both font tar-balls from here.

Now that the fonts are in place go back to your home directory. A few files need patching, see patches below.

```
tar -zxvf /usr/src/ghostscript/ghostscript-6.50.tar.gz
cd gs6.50
tar -zxvf /usr/src/ghostscript/ghostscript-6.50jpeg.tar.gz
```

```
tar -zxvf /usr/src/ghostscript/ghostscript-6.50libpng.tar.gz
tar -zxvf /usr/src/ghostscript/ghostscript-6.50zlib.tar.gz
ln -s libpng-1.0.8 libpng
ln -s zlib-1.1.3 zlib
ln -s jpeg-6b jpeg
ln -s src/unix-gcc.mak Makefile
# now edit Makefile, lib/lprsetup.sh, lib/unix-lpr.sh; see patch below
make
make install
strip /usr/local/ghostscript/bin/gs
gzip -9 /usr/local/ghostscript/man/man1/*
# now fix the symbolic links in the man directory
```

Setting up the postscript printer driver

Ghostscript comes with a filter that allows printing of postscript files on an ordinary line printer (at least if you have a supported one). `Cd /usr/local/share/ghostscript` and look at `lprsetup.sh`. It should be modified to list your printer as the device, and your spool directory as the spool directory. The patch which I wrote should change this correctly (see the listing of the patch below). If everything is alright run the script. This creates the filter directory and a few links in it, and a file *printcap.insert* which hopefully is going to work with your printer. It needs to be inserted into */etc/printcap* and *modified*. I use a parallel printer */dev/lp0* and keep the spool files in */var/spool/lp0*. Create the files *.seq*, *logfile*, *acct*, *lock*, *errs*, *status* in */var/spool/lp0* and set permissions and ownership to `--rw-rw-r-- root lp`. You might already have set up some of those files if you set up the printer to print text files earlier.

A printcap entry

```
bj200ps|ghostscriptprinter:\
:lp=/dev/lp0:\
:ms=-parity,ixon,-opost:\
:sd=/var/spool/lp0:\
:lf=/var/spool/lp0/logfile:\
:af=/var/spool/lp0/acct:\
:if=/usr/local/ghostscript/share/ghostscript/filt/direct/bj200/gsif:\
:mx#0:sf:sh:rs:
```

The structure of /usr/local/ghostscript/share/ghostscript/filt

```
lrwxrwxrwx   1 root    root     1 Feb  6 19:38 indirect -> .
lrwxrwxrwx   1 root    root    14 Feb  6 19:38 gsif -> ../../../bin/unix-lpr.sh
lrwxrwxrwx   1 root    root     1 Feb  6 19:38 direct -> .
lrwxrwxrwx   1 root    root     1 Feb  6 19:38 deskjet -> .
lrwxrwxrwx   1 root    root     1 Feb  6 19:38 bj200 -> .
```

The contents of the patch file

```
diff -rc gs6.50/lib/lprsetup.sh gs6.50-good/lib/lprsetup.sh
*** gs6.50/lib/lprsetup.sh Thu Mar  9 00:40:40 2000
--- gs6.50-good/lib/lprsetup.sh Tue Feb  6 20:02:12 2001
```

```
***************
*** 1,23 ****
  #!/bin/sh
- # $Id: lprsetup.sh,v 1.1 2000/03/09 08:40:40 lpd Exp $
  #
  # BSD PRINT FILTER SETUP utility for Ghostscript - used and tested on
  # SunOS 4.1.3, but I hope it will be useful on other BSD systems
  # See documentation for usage
  #

! DEVICES="bjt600.32 bjc600.32 bjc600.24 bjc600.24.3 bjc600.16 bjc600.8 bjc60
0.8.1 bjc600.1 bjc600.dq"
  #FILTERS="if nf tf gf vf df cf rf"
  FILTERS="if"

  # The port your printer is on
! PRINTERDEV=/dev/lp1
  # The kind of printer (accepted values: 'parallel' and 'serial')
  PRINTERTYPE=parallel

! GSDIR=/usr/local/lib/ghostscript
  GSFILTERDIR=$GSDIR/filt
! SPOOLDIR=/var/spool
  GSIF=unix-lpr.sh
  PCAP=printcap.insert

--- 1,23 ----
  #!/bin/sh
  #
  # BSD PRINT FILTER SETUP utility for Ghostscript - used and tested on
  # SunOS 4.1.3, but I hope it will be useful on other BSD systems
  # See documentation for usage
  #

! #DEVICES="bjt600.32 bjc600.32 bjc600.24 bjc600.24.3 bjc600.16 bjc600.8 bjc60
0.8.1 bjc600.1 bjc600.dq"
! DEVICES="bj200 deskjet"
  #FILTERS="if nf tf gf vf df cf rf"
  FILTERS="if"

  # The port your printer is on
! PRINTERDEV=/dev/lp0
  # The kind of printer (accepted values: 'parallel' and 'serial')
  PRINTERTYPE=parallel

! GSDIR=/usr/local/share/ghostscript
  GSFILTERDIR=$GSDIR/filt
! SPOOLDIR=/var/spool/lp0
  GSIF=unix-lpr.sh
  PCAP=printcap.insert

diff -rc gs6.50/lib/unix-lpr.sh gs6.50-good/lib/unix-lpr.sh
```

```
*** gs6.50/lib/unix-lpr.sh Thu Mar  9 00:40:40 2000
--- gs6.50-good/lib/unix-lpr.sh Tue Feb  6 20:02:02 2001
***************
*** 1,5 ****
  #!/bin/sh
- # $Id: unix-lpr.sh,v 1.1 2000/03/09 08:40:40 lpd Exp $
  #
  # Unix lpr filter. The default setup sends output directly to a pipe,
  # which requires the Ghostscript process to fork, and thus may cause
--- 1,4 ----
***************
*** 16,27 ****
  # 'gsoutput' and uncommenting the lines referring to 'gspipe'.
  #

! PBMPLUSPATH=/usr/local/bin
! PSFILTERPATH=/usr/local/lib/ghostscript
  LOCALPATH=/usr/local/bin
! X11HOME=/usr/X11R6

! PATH=/bin:/usr/bin:/usr/ucb:/usr/etc
  PATH=${PATH}\:${LOCALPATH}\:${PBMPLUSPATH}\:${PSFILTERPATH}
  LD_LIBRARY_PATH=${X11HOME}/lib

--- 15,26 ----
  # 'gsoutput' and uncommenting the lines referring to 'gspipe'.
  #

! PBMPLUSPATH=
! PSFILTERPATH=/usr/local/share/ghostscript
  LOCALPATH=/usr/local/bin
! X11HOME=/usr/X11

! PATH=/bin:/usr/bin:/usr/ucb:/usr/X11/bin
  PATH=${PATH}\:${LOCALPATH}\:${PBMPLUSPATH}\:${PSFILTERPATH}
  LD_LIBRARY_PATH=${X11HOME}/lib

***************
*** 62,84 ****
  #
  # Find the bpp and number of colors, if specified
  #

! colorspec="`echo ${device} | sed 's/.*\.[0-9][0-9]*\.\([0-9][0-9]*\)$/\1/'`"
! if test "$colorspec" = "${device}"
! then
!     colorspec=""
! else
!     device=`basename ${device} .$colorspec`
!     colorspec="-dColors=$colorspec"
! fi
!
```

```
! bpp="‘echo ${device} | sed ’s/.*\.\([0-9][0-9]*\)$/\1/’‘"
! if test "$bpp" = "${device}"
! then
!     bpp=1
! else
!     device=‘basename ${device} .$bpp‘
! fi

  #
  # Information for the logfile
--- 61,86 ----
  #
  # Find the bpp and number of colors, if specified
  #
+ #
+ # colorspec="‘echo ${device} | sed ’s/.*\.[0-9][0-9]*\.\([0-9][0-9]*\)$/\1/’‘"
+ # if test "$colorspec" = "${device}"
+ # then
+ #     colorspec=""
+ # else
+ #     device=‘basename ${device} .$colorspec‘
+ #     colorspec="-dColors=$colorspec"
+ # fi
+ #
+ # bpp="‘echo ${device} | sed ’s/.*\.\([0-9][0-9]*\)$/\1/’‘"
+ # if test "$bppspec" = "${device}"
+ # then
+ #     bpp=1
+ # else
+ #     device=‘basename ${device} .$bpp‘
+ # fi

! bpp=1
! colorspec=""

  #
  # Information for the logfile
***************
*** 88,94 ****
  job=‘egrep ’^J’ ${cf} | tail +2c‘

  echo "gsbanner: ${host}:${user}  Job: ${job}  Date: ‘date‘"
! echo "gsif: ${host}:${user} ${fdevname} start - ‘date‘"

  #
  # Set the direct or indirect output destinations
--- 90,96 ----
  job=‘egrep ’^J’ ${cf} | tail +2c‘

  echo "gsbanner: ${host}:${user}  Job: ${job}  Date: ‘date‘"
! # echo "gsif: ${host}:${user} ${fdevname} start - ‘date‘"
```

```
  #
  # Set the direct or indirect output destinations
***************
*** 127,159 ****
  #
  # This is the postlude which does the accounting
  #
! echo "\
! (acctfile) getenv
!   { currentdevice /PageCount gsgetdeviceprop dup cvi 0 gt
!     { exch (a) file /acctfile exch def
!       /string 20 string def
!       string cvs dup length dup
!       4 lt
!         { 4 exch sub
!           { acctfile ( ) writestring } repeat
!         } { pop } ifelse
!       acctfile exch writestring
!       acctfile (.00 ) writestring
!       acctfile (host) getenv
!         { string cvs } { (NOHOST) } ifelse writestring
!       acctfile (:) writestring
!       acctfile (user) getenv
!         { string cvs } { (NOUSER) } ifelse writestring
!       acctfile (\n) writestring
!       acctfile closefile
!     } { pop } ifelse
!   } if
! quit"
  ) | gs -q -dNOPAUSE -sDEVICE=${device} -dBitsPerPixel=${bpp} $colorspec \
   -sOutputFile=\|"${gsoutput}" -
  # -sOutputFile=${gspipe} -

! rm -f ${gspipe}
  #
  # End the logfile entry
  #
--- 129,161 ----
  #
  # This is the postlude which does the accounting
  #
! # echo "\
! # (acctfile) getenv
! #   { currentdevice /PageCount gsgetdeviceprop dup cvi 0 gt
! #     { exch (a) file /acctfile exch def
! #  /string 20 string def
! #  string cvs dup length dup
! #  4 lt
! #    { 4 exch sub
! #      { acctfile ( ) writestring } repeat
! #    } { pop } ifelse
! #  acctfile exch writestring
```

```
! #  acctfile (.00 ) writestring
! #  acctfile (host) getenv
! #    { string cvs } { (NOHOST) } ifelse writestring
! #  acctfile (:) writestring
! #  acctfile (user) getenv
! #    { string cvs } { (NOUSER) } ifelse writestring
! #  acctfile (\n) writestring
! #  acctfile closefile
! #     } { pop } ifelse
! #   } if
! # quit"
  ) | gs -q -dNOPAUSE -sDEVICE=${device} -dBitsPerPixel=${bpp} $colorspec \
   -sOutputFile=\|"${gsoutput}" -
  # -sOutputFile=${gspipe} -

! # rm -f ${gspipe}
  #
  # End the logfile entry
  #
diff -rc gs6.50/src/unix-gcc.mak gs6.50-good/src/unix-gcc.mak
*** gs6.50/src/unix-gcc.mak Mon Sep 25 08:06:28 2000
--- gs6.50-good/src/unix-gcc.mak Tue Feb  6 19:33:33 2001
***************
*** 56,62 ****
  INSTALL_PROGRAM = $(INSTALL) -m 755
  INSTALL_DATA = $(INSTALL) -m 644

! prefix = /usr/local
  exec_prefix = $(prefix)
  bindir = $(exec_prefix)/bin
  scriptdir = $(bindir)
--- 56,62 ----
  INSTALL_PROGRAM = $(INSTALL) -m 755
  INSTALL_DATA = $(INSTALL) -m 644

! prefix = /usr/local/ghostscript
  exec_prefix = $(prefix)
  bindir = $(exec_prefix)/bin
  scriptdir = $(bindir)
***************
*** 254,260 ****
  # Note that x_.h expects to find the header files in $(XINCLUDE)/X11,
  # not in $(XINCLUDE).

! XINCLUDE=-I/usr/local/X/include

  # Define the directory/ies and library names for the X11 library files.
  # XLIBDIRS is for ld and should include -L; XLIBDIR is for LD_RUN_PATH
--- 254,260 ----
  # Note that x_.h expects to find the header files in $(XINCLUDE)/X11,
  # not in $(XINCLUDE).
```

```
! XINCLUDE=-I/usr/X11/include

  # Define the directory/ies and library names for the X11 library files.
  # XLIBDIRS is for ld and should include -L; XLIBDIR is for LD_RUN_PATH
```

42 Ghostview

Files:
/usr/X11R6/bin/{gv, ghostview}
*/usr/X11R6/lib/gv/**
/usr/X11R6/man/man1/{gv.1.gz, ghostview.1.gz}
/usr/X11R6/lib/X11/app-defaults/GV

Sources obtainable at:
`http://wwwthep.physik.uni-mainz.de/ plass/gv/`
`ftp://thep.physik.uni-mainz.de/pub/gv/`

Edit the file *config.Unix* and say that the files should go into */usr/X11/**. Then

```
xmkmf
make Makefiles
make
make install
cd doc
mv gv.man gv.1
gzip -9 man.1
install -m 644 gc.1.gz /usr/X11/man/man1/
cd /usr/X11/bin
strip gv
ln -s gv ghostview
cd /usr/X11/man/man1/
ln -s gv.1.gz ghostview.1.gz
```

43 GNU Compilers: gcc, g++, g77, et al.

gcc
Files:
/usr/bin/{c++, g++, gcc, protoize, unprotoize, cpp}
/lib/cpp
/usr/i486-pc-linux-gnulibc1/include/
*/usr/lib/gcc-lib/i486-pc-linux-gnulibc1/2.8.1/**
/usr/man/man1/{cccp.1.gz, g++.1.gz, gcc.1.gz}
/usr/info/{cpp.info, gcc.info*}*

libstdc++

Files:
/usr/lib/{libiberty.a,libstdc++.a}
/usr/i486-pc-linux-gnulibc1/include/_G_config.h
/usr/i486-pc-linux-gnulibc1/lib/libiberty.a
*/usr/include/g++/**

Sources obtainable at:
`ftp://ftp.gnu.ai.mit.edu/pub/gnu/libstdc++-2.8.1.tar.gz`
`ftp://ftp.gnu.ai.mit.edu/pub/gnu/gcc-2.8.1.tar.gz`

The old installation for `aout` is still there as well. You can run the old version with `gcc -V 2.7.2 -b i486-linuxaout`. I removed */usr/i486-linux* and */usr/lib/gcc-lib/i486-linux/2.7.2* after I had moved the *include* subdirectory from that last directory to */usr/lib/gcc-lib/i486-linuxaout/2.7.2/*.
aout
Files:
*/usr/i486-linuxaout/**
*/usr/lib/gcc-lib/i486-linuxaout/2.7.2/**

old libg++
Files:
/usr/info/{libg++.info, iostream.info*, configure.info*, standards.info*, cfg-paper.info*, g++FAQ.info*, gperf.info*}*
/usr/bin/{genclass, gperf}
/usr/man/man1/gperf.1.gz

43.1 Unpacking the sources and such

First unpack `libstdc++` and then change to the newly created directory and unpack `gcc` from there. If so desired one can also unpack a copy of `egcs` to get the test suite. This will then have to be moved into the directory *gcc*. Finally, there is patch in the *INSTALL* file of libstdc++ which allows to use the testsuite. So here it goes.

```
tar -zxvf libstdc++-2.8.1.tar.gz
cd libstdc++-2.8.1
tar -zxvf ../gcc-2.8.1.tar.gz
ln -s gcc-2.8.1 gcc
bzip2 -dc ../egcs-1.0.2.tar.bz2 | tar -xvf -
mv egcs-1.0.2/gcc/testsuite gcc
rm -rf egcs-1.0.2
patch -p0 <INSTALL
cd ..
mkdir built
cd built
CFLAGS="-O2 -m486" ../libstdc++-2.8.1/configure --prefix=/usr
make
make check
make install
```

Then compress the man pages and the info pages. I also removed */usr/bin/i486-pc-linux-gnulibc1-gcc* because it is just a copy of *gcc*.

Gcc with g77, the old-fashioned way

Do not install this way as g77 breaks gcc! Install gcc alone only. Also, gcc is now at version 2.8.1 and g77 version 0.5.22 (the latest) cannot be installed with that newest version of gcc. The last version of gcc to compile with g77 is gcc version 2.7.3.2. The GNU C-compiler and Fortran compiler are available from any GNU site, in particular from `ftp://ftp.gnu.ai.mit.edu`. Unpacking the sources involves several steps. First unpack the *gcc* sources, then apply any upgrade patches, then unpack the sl g77 sources, move them into the *gcc subdirectory*, and finally apply the patch that comes with the *g77* sources. Then reformat the info pages as one of the patches breaks them.

```
tar -zxf gcc-2.7.2.1.tar.gz
cd gcc-2.7.2.1
zcat ../gcc-2.7.2.1-2.7.2.2.diff.gz | patch -p0
zcat ../gcc-2.7.2.2-2.7.2.3.diff.gz | patch -p1
cd ..
mv gcc-2.7.2.1 gcc
tar -zxf g77-0.5.20.tar.gz
mv g77-0.5.20/* gcc/
cd gcc
patch -p1 < f/gbe/2.7.2.2.diff
make -f Makefile.in gcc.info
```

Gcc alone, no libstdc++, the old-fashioned way

Just unpack the sources with `tar -zxf gcc-2.8.1.tar.gz`, apply any patches, and change the directory.

Compilation

Now you are ready to go. Use the old compiler to compile a basic version, then use this new compiler to compile the real compiler, and then use this latter one to compile it again, and compare if you got the same binaries. This is a built-in safety measure. Finally install. The configure step below will not work if you try to configure for `i486-linux`. I also tried to compile a cross compiler for aout binaries, but it never worked. Probably my mistake.

```
# before version 2.7.2.3 it was
#./configure --build=i486-linuxlibc1 --prefix=/usr\
#        --local-prefix=/usr/local --gxx-include-dir=/usr/include/g++
# for version 2.7.2.3 it was
# ./configure --build=i486-linux-gnulibc1 --prefix=/usr\
#        --local-prefix=/usr/local --gxx-include-dir=/usr/include/g++
# for version 2.8.1 it is just
./configure --prefix=/usr
# touch f2c-install-ok # if you install f2c
# touch f77-install-ok # if you want f77
make LANGUAGES=c
make stage1
make CC="stage1/xgcc -Bstage1/" CFLAGS="-O2 -m486" LANGUAGES="c"
make stage2
```

```
make CC="stage2/xgcc -Bstage2/" CFLAGS="-O2 -m486" LANGUAGES="c"
make compare
make CC="stage2/xgcc -Bstage2/" CFLAGS="-O2 -m486"\
LANGUAGES="c c++ objective-c proto"
# if you want f77 and g77 add them both to the LANGUAGES above
make install CC="stage2/xgcc -Bstage2/" CFLAGS="-O2 -m486"\
LANGUAGES="c c++ objective-c proto"
# if you want f77 and g77 add them both to the LANGUAGES above
cd /usr/bin
rm cpp
ln -s /usr/lib/gcc-lib/i486-pc-linux-gnulibc1/2.8.1/cpp
cd /lib
rm cpp
ln -s /usr/lib/gcc-lib/i486-linux-gnulibc1/2.8.1/cpp
# if you have g77 then ln -s /usr/lib/gcc-lib/i486-linux-gnulibc1/2.8.1.f/cpp
```

Finally edit the file */usr/info/dir* to add the new info pages, then compress the info and the man pages and strip the executables. I also remove the binary */usr/bin/i486-pc-linux-gnulibc1-gcc* because it is identical to */usr/bin/gcc*. If you have libg++ installed then make sure you have the link */usr/lib/g++-include —> /usr/include/g++* because *g++* will look there for its include files.

44 GNU Debugger

Files:
/usr/bin/gdb
*/usr/doc/gdb/**
/usr/man/man1/gdb.1.gz
/usr/info/{gdb.info,gdbint.info*,stabs.info*}*

Unpack the source, which contains lots of other stuff, too. `Cd` into the directory that will be created, and run the following commands.

```
./configure
make
cd gdb
make install
make install-info
cd doc
gzip -9 *.dvi
install -m 644 *.dvi.gz /usr/doc/gdb
strip /usr/bin/gdb
gzip -9 /usr/man/man1/gdb.1 /usr/info/gdb.* /usr/info/gdbint.* /usr/info/stabs.*
vi /usr/info/dir               # Now add entries for the new info pages.
```

45 Glibc2

Files:
*/glibc2/**

*/usr/i686-pc-linux-gnulibc2/lib/**
/etc/ld.so.conf
/lib/ld-linux.so.2

Sources obtainable at:
`http://ftp.gnu.ai.mit.edu/pub/gnu/glibc`

Installation instructions are below, also read the files NOTES and INSTALL that come with the sources. Definitely read about upgrading the C-library in Section . I installed only the shared libraries, none of the other development tools or header files or whatever. In fact, I ultimately down-loaded a collection of shared libraries from *gwyn.tux.org*, which is a gnulibc2-based system, and just installed those. Make sure that *ld-linux.so.2* is found in */lib*, because that location is hard-coded into the binaries. For me this is simply a symbolic link to the real thing.

```
tar -zxvf /usr/src/gnu/glibc/glibc-2.1.2.tar.gz
cd glibc-2.1.2
tar -zxvf /usr/src/gnu/glibc/glibc-linuxthreads-2.1.2.tar.gz
tar -zxvf /usr/src/gnu/glibc/glibc-crypt-2.1.2.tar.gz
cd ..
mkdir build
cd build
../glibc-2.1.2/configure \
--prefix=/usr \
--disable-nls \
--with-headers=/usr/src/linux/include \
--enable-add-ons
make
make check
su
# now first install elsewhere to make a binary distribution
make install install_root=/home/tmp/glibc
cd /home/tmp/glibc
strip sbin/* usr/bin/* usr/sbin/* usr/libexec/*
gzip -9 usr/info/*
tar -zcvf ../glibc-2.1.2-i686-bin.tar.gz *
```

Some notes about compileing times and disk use.

```
Disk use of sources in KBytes:                      43152
Disk use of build tree in KBytes:                  238543
Disk use of build tree in KBytes after make check: 253779
Disk use of installation, after stripping binaries
and compressing info pages:                         85867

Time use running make:        real: 50m44s   user: 36m46s   sys: 6m11s
Time use running make check:  real:  7m25s   user:  5m 7s   sys: 0m21s
*These times are on a Pentium-II 266 MHz with 96MB RAM.
```

46 Gnuplot

Files:

*/usr/local/gnuplot/**

Sources obtainable at:
ftp://ftp.gnuplot.vt.edu/pub/gnuplot/gnuplot-3.7.tar.gz
http://members.theglobe.com/gnuplot/gp37tar.gz
http://www.geocities.com/SiliconValley/Foothills/6647/gp37tar.gz
ftp://ftp.dartmouth.edu/pub/gnuplot/gnuplot-3.7.tar.gz

```
CFLAGS="-O2 -fomit-frame-pointer" ./configure \
 --without-readline\
 --with-gnu-readline\
 --without-gd\
 --enable-system-time\
 --without-linux-vga\
 --prefix=/usr/local/gnuplot
make
# testing, the "hash -r" assumes you are running bash
PATH=‘pwd‘:$PATH
hash -r
cd demo
gnuplot all.dem </dev/null
# a much shorter test is obtained by simply starting gnuplot and then issuing
# the command "test".
# end of testing
make install
# optional installing of documentation
mkdir /usr/local/gnuplot/docs /usr/local/gnuplot/demo
install -m 644 demo/* /usr/local/gnuplot/demo/
install -m 644 0FAQ /usr/local/gnuplot/docs/FAQ.txt
cd docs
make html
sed -e 's|src="http://[^"]*/\([^/]*\)"|src="\1"|g' gnuplot.html > \
  /usr/local/gnuplot/docs/gnuplot.html
tex gpcard
install -m 644 gpcard.dvi /usr/local/gnuplot/docs/
cd latextut
make
install -m 644 tutorial.dvi /usr/local/gnuplot/docs/LaTeX-tutorial.dvi
```

As always, strip the binaries, compress man, HTML-formatted, and text documentation pages, and link the whole thing from the corresponding places in */usr/local/*. The HTML-formatted documentation tries to obtain a few graphics from the Internet. The solution is to either obtain those once and store them locally, or to simply not getting them. On any case the file needs to be modified, or Netscape will hang trying to obtain the pictures, while it is off-line. The command is given above in the install instructions.

47 HTML Formatted Documentation

Files:
*/usr/local/html/**

These are documentation files which can be read using Netscape, for example. Any other documentation files go into */usr/local/doc/*. The top node of the HTML documentation tree is the file */usr/local/html/index.html*. Modify this file to include new files.

index.html

```
<html>
<title>
Local documentation
</title>
<body>
<h2>
Local documentation
</h2>
<p>
<dt> <a href="file:/usr/local/MuPAD/WWW/index.html">MuPAD</a>
<dt> <a href="file:/usr/doc/teTeX/helpindex.html">teTeX</a>
<dt> <a href="file:/usr/local/html/troff_summary.html">troff</a>
<dt> <a href="file:/usr/local/html/groff_toc.html">groff</a>
<hr>
<dt> <a href="http:/usr/local/doc/">other formats</a>
<hr>
<i> Last updated 02/12/96 </i>
</body>
</html>
```

48 HTML to PDF converter

Files:
/usr/bin/htmldoc
*/usr/share/htmldoc/**
/usr/man/man1/htmldoc.1.gz
/etc/software/htmldoc.remove

Sources obtainable at:
`ftp://ftp.funet.fi/pub/mirrors/ftp.easysw.com`

Install by running the *htmldoc.install* script. Then compress the man page. If this is an update, remove the old software first by running the removal script in */etc/software*, because the program will make a copy of the old software in place with *.O* extensions.

49 ImageMagick

Files:
*/usr/local/magick/**

Sources obtainable at:
`http://www.wizards.dupont.com/cristy/ImageMagick.html`
`ftp://ftp.wizards.dupont.com/pub/ImageMagick/`

The package also needs jpeg, tiff, mpeg, png, and zlib libraries installed. Unpack sources, and run

```
CFLAGS="-O2" ./configure --disable-static --prefix=/usr/local/magick
make
mkdir /usr/local/magick
make install
```

Compress the man pages, strip the binaries, link everything from the correct place in */usr/local*.

50 Info Files

Files:
*/usr/info/**

These are info files which can be read using `info` or `emacs`. Any other info files go into the same directory. The top node of the info tree is */usr/info/dir*. Modify this file to include new info files. Info files come with tag lists in them that tell `info` where the nodes are. I wrote a script that roughly recreates such a list. It reads from standard input and writes to standard output. It only needs to be run if you get an info file with an out-of-date tag list.

maketags.perl

```
#!/usr/bin/perl
# This script expects an info-formatted file on standard input and will
# write a list of its nodes (tags) to standard output.
# Uwe F. Mayer, Feb 11, 1996.
$position=0;
while ($line=<STDIN>) {
     if ($line=~/^{ }_/) {
    $tag=$position;
     }
     $position=$position+length($line);
     if ($line=~/Node:/) {
 chop($line);
 $line=~s/.*(Node:.*)/$1/;
 $line=~s/,.*$//;
 printf("$line^{ }?$tag\n");
     }
}
```

51 Ispell

Files:
/usr/bin/{ispell buildhash icombine ijoin munchlist findaffix tryaffix sq unsq}
/usr/man/man1/{ispell.1.gz sq.1.gz buildhash.1.gz munchlist.1.gz findaffix.1.gz tryaffix.1.gz}
/usr/man/man4/ispell.4.gz
/usr/share/emacs/site-lisp/{ispell.el ispell.elc}
*/usr/lib/ispell/**
/usr/info/ispell.info.gz

Sources obtainable at:
`http://fmg-www.cs.ucla.edu/fmg-members/geoff/ispell.html`
`http://www.nyx.net/~cmaarbj/ispell/` for the html patch

The first is the home page of *ispell*, the second is the place where a patch to process HTML formatted documents can be found. Unpack the sources, `cd` to the generated directory and apply the patch from here. Then `cp local.h-samp local.h`. Now edit this file.

```
line 84:  #define USG             /* Define this on System V */
line 89:  #define BINDIR          "/usr/bin"
line 90:  #define LIBDIR          "/usr/lib/ispell"
line 91:  #define ELISPDIR        "/usr/share/emacs/site-lisp"
line 92:  #define TEXINFODIR      "/usr/info"
line 93:  #define MAN1DIR         "/usr/man/man1"
line 94:  #define MAN4DIR         "/usr/man/man4"
line 99:  #define CC              "gcc"
line 100: #define CFLAGS          "-O6 -funroll-loops"
line 101: #undef NO8BIT
line 102: #define HTSPECIAL
line 103: #define MASKBITS        64
line 104: #define MASKTYPE        int
line 105: #define MASKTYPE_WIDTH  32
```

Note that the last three lines have to do with how many affix rules you may have. Setting `MASKBITS` to 32, which is the default, allows for 26 such rules. However, some languages have more (for example Greek), and setting it to 64 allows for 58 of them. The details are in *config.X*. `MASKTYPE` should be a type that the processor can access fast. Now finally type `export TMPDIR=/usr/tmp` and then `make all` and `make install`. During the `make install` step there is an error when byte-compiling the Emacs addon.

51.1 Other languages for ispell

51.1.1 British

The default of ispell is American style English. If you want at British dictionary as well, then you can build one when from the sources of ispell.

```
cd ispell-3.1/languages/english
munchlist -v -l english.aff /usr/dict/words english.0 english.1 british.0 britis
h.1 >britishmed+
buildhash britishmed+ english.aff britishmed+.hash
```

51.1.2 German

The sources for the german dictionary are available via anonymous ftp from *ftp.informatik.uni-kiel.de/pub/kiel/dicts/hk2-deutsch.tar.gz*. Unpack the sources into an empty directory, `tar` does not create a subdirectory for the files. For a standard dictionary run

```
cat {worte,verben,adjektive,klein,geographie,vornamen,abkuerz,imperat,latein,info
rmatik,infoabk}.txt | sort -u > all.words
buildhash all.words deutsch.aff deutsch.hash
```

51.1.3 Spanish

The spanish dictionary is available at `ftp.fi.upm.es/pub/unix/espa~nol.tar.gz`. Unpack the sources and cd into the directory. Then run `make e~ne` if you want to specify `~n` instead of `'n` for the letter `e~ne`. Then run `make`. Takes quite a while. If you run out of disk space, then set `export TMPDIR=/usr/tmp`, or wherever you have lots of space.

51.1.4 French

For the french dictionary it is even simpler. Get the french sources from ispell's homepage, unpack them into an empty directory, and run

```
buildhash francais.dico ./francais.aff francais.hash
```

51.1.5 For all of them

Install the files **.hash* into */usr/lib/ispell*. You choose a different language by specifying the `-d` flag for `ispell`, so for example `ispell -d deutsch <filename>` gets the German language going. From inside Emacs first change the dictionary with (`M-x ispell-change-dictionary deutsch`) and then use it as usual with (`M-x ispell-buffer` or `M-x ispell-region`).

However, Spanish is not one of the languages set up for use with ispell from within Emacs, and so it needs to be added to the list of available dictionaries. This is done by adding a few entries to the variable `ispell-dictionary-alist` which is set in the file *ispell.el*. However, you do not have to edit *ispell.el* but you simply set this variable from inside of *default.el*. The syntax seems to have changed since that entry below has been produced, but here it is anyways.

```
(setq ispell-dictionary-alist '( ... copy all entries from ispell.el ...
  ("espa~nol" "[A-Za-z]" "[^A-Za-z]" "[---'~\"]" t nil nil)
  ("espa~nol8"
     "[A-Z\301\311\315\323\332\334\321a-z\341\351\355\363\372\374\361]"
     "[^A-Z\301\311\315\323\332\334\321a-z\341\351\355\363\372\374\361]"
     "[---]" nil ("-T" "latin1" "-d" "espa~nol") nil)
  ("espa~nol-tex" "[A-Za-z]" "[^A-Za-z]" "[---'~\\\"]" t
     ("-T" "TeX" "-d" "espa~nol") nil)))
```

The new way around is, to link the file *espa~nol.hash* to *castellano.hash*, and one uses now `castellano` or `castellano8` from within Emacs.

52 Java

Files:
/usr/local/jdk
/usr/local/j2sdk1.3.1

Sources obtainable at:
`http://www.blackdown.org/java-linux.html`
`http://java.sun.com/j2se/1.3/docs.html`

The Java port to Linux comes in two files, one containing the developers kit and one the documentation. Untar the development kit from */usr/local* and adjust the path in */etc/profile* to contain */usr/local/jdk/bin*. I keep a symbolic link from *jdk* to *j2sdk1.3.1*. The development kit is available from blackdown.org, while the documentation (125MB!) is available directly from Sun. The documentation is installed from within the jdk directory. Make an entry in the index in */usr/local/html*.

The standard symbol fonts can have different names on different Linux systems. I had to edit the file *jre/lib/font.properties* and had to replace "standard symbols l" with "symbol" in the font names. This is also explained in the header of that file.

53 KDE

Files:
/usr/local/kde
/etc/profile
/etc/ld.so.conf

Sources obtainable at:
`http://www.kde.org`

You need to install QT (see Section 82.2) version 1.44 first to get kde version 1.1.2 to compile.

```
#*************** Important ************************
#
# add /usr/local/kde/bin to your PATH
# add /usr/local/kde/bin to /etc/ld.so.conf

bunzip2 < /usr/src/kde/kdelibs-1.1.2.tar.bz2 | tar -xvf -
cd kdelibs-1.1.2
./configure --disable-nls
make
make check
make install
cd ..

bunzip2 < /usr/src/kde/kdesupport-1.1.2.tar.bz2| tar -xvf -
cd kdesupport-1.1.2
# I already have libjpeg and libgdbm
rm -rf jpeglib6a/
rm -rf gdbm/
./configure --disable-nls --without-libjpeg --without-libgdbm --disable-static
make
make check
make install
cd ..

bunzip2 < /usr/src/kde/kdebase-1.1.2.tar.bz2 | tar -xvf -
##
## I get a few errors without the following:
## Edit the file kvt/utmp.c and add a line with "#undef UTMP_SUPPORT"
## Edit the file kdm/session.c and comment out the definition of crypt on
```

```
##      lines 1093-1097
##
cd kdebase-1.1.2
./configure --disable-nls
make
make check
make install
cd ..

bunzip2 < /usr/src/kde/kdeadmin-1.1.2.tar.bz2 | tar -xvf -
cd kdeadmin-1.1.2
./configure --disable-nls
make
make check
make install
cd ..

bunzip2 < /usr/src/kde/kdeutils-1.1.2.tar.bz2 | tar -xvf -
cd kdeutils-1.1.2
./configure --disable-nls
make
make check
make install
cd ..

bunzip2 < /usr/src/kde/kdenetwork-1.1.2.tar.bz2 | tar -xvf -
cd kdenetwork-1.1.2
./configure --disable-nls
make
make check
make install
cd ..

bunzip2 < /usr/src/kde/kdegraphics-1.1.2.tar.bz2 | tar -xvf -
cd kdegraphics-1.1.2
./configure --disable-nls
make
make check
make install
cd ..

bunzip2 < /usr/src/kde/kdemultimedia-1.1.2.tar.bz2 | tar -xvf -
cd kdemultimedia-1.1.2
./configure --disable-nls
make
make check
make install
cd ..

bunzip2 < /usr/src/kde/kdegames-1.1.2.tar.bz2 | tar -xvf -
cd kdegames-1.1.2
./configure --disable-nls
```

```
make
make check
make install
cd ..

bunzip2 < /usr/src/kde/kdetoys-1.1.2.tar.bz2 | tar -xvf -
cd kdetoys-1.1.2
./configure --disable-nls
make
make check
make install

strip /usr/local/kde/bin/*

# In /usr/local/kde/share/doc/HTML remove all but the 'default'
# and the 'en' directories
```

Compilation and installation takes a while, after all this is a 42 MB installation when it's all said and done. The times below are for a Pentium II 266 MHz processor with 96 MB RAM and light system load.

13 minutes for kdelibs-1.1.2
6 minutes for kdesupport-1.1.2
22 minutes for kdebase-1.1.2
5 minutes for kdeadmin-1.1.2
9 minutes for kdeutils-1.1.2
20 minutes for kdenetwork-1.1.2
9 minutes for kdegraphics-1.1.2
5 minutes for kdemultimedia-1.1.2
6 minutes for kdegames-1.1.2
2 minutes for kdetoys-1.1.2

54 Kermit

Files:
/usr/local/bin/kermit
/usr/local/man/cat1/kermit.1.gz
*/usr/local/lib/kermit/**

Sources obtainable at:
`http://www.columbia.edu/kermit`

Unpacking the source archive does not create a subdirectory, so make one yourself, say, *ckermit*. Compile with

```
make linux KFLAGS="-DNODEBUG -I/usr/include/ncurses"
```

The resulting binary is called *wermit*, strip and install it. Then, as usual, fix the permissions and owner of the man page (called *ckuker.nr*), look at the man page, and remove the unformatted version if you don't want to keep it.

Install the documentation into */usr/local/lib/kermit*, which files to keep is explained in the file *ckuins.doc*.

55 Lacheck

Files:
/usr/local/bin/lacheck
/usr/local/man/cat1/lacheck.1.gz

Sources obtainable at:
`http://www.cl.cam.ac.uk/tex-archive/support/lacheck/lacheck.tar.gz`

Unpack the source, this creates a subdirectory with the files in it. Apply the patch to configure the Makefile: `gzip -dc /mnt/lach_pat.gz | patch -p0`. Become root and `cd` to *lacheck-1.2.1*. Run `make uninstall` if you have an old copy, then run `make install`.

Makefile changes

```
diff -cr lacheck-1.21~/Makefile lacheck-1.21/Makefile
*** lacheck-1.21~/Makefile      Wed Aug 30 13:12:59 1995
--- lacheck-1.21/Makefile       Thu Feb  1 23:26:33 1996
***************
*** 7,22 ****
  exec_prefix = $(prefix)

  bindir = $(exec_prefix)/bin
! mandir = $(prefix)/man/man1
! manext = .1
  srcdir = .

  # Use 'cc -bsd' on a next
  # use 'cc -cckr' on a sgi
  # CC = gcc -traditional
! CC=cc

! CFLAGS = -g # -O -DNEED_STRSTR

  # No changes should be needed below

--- 7,22 ----
  exec_prefix = $(prefix)

  bindir = $(exec_prefix)/bin
! mandir = $(prefix)/man/cat1
! manext = .1.gz
  srcdir = .

  # Use 'cc -bsd' on a next
  # use 'cc -cckr' on a sgi
  # CC = gcc -traditional
! CC=gcc
```

```
! CFLAGS = -g -O2 # -DNEED_STRSTR

  # No changes should be needed below

***************
*** 56,62 ****
  lacheck.1:    lacheck.man
        -sed -e "s!%%LACHECKREV%%!Release $(REV)!" $(srcdir)/lacheck.man |\
        sed -e "s!%%LACHECKDATE%%!`date +%D`!" |\
!       sed -e "s!%%LACHECKPATH%%!$(bindir)/$(LACHECK)!" > $@

  test.new:     test.tex $(LACHECK)
        $(srcdir)/$(LACHECK) $(srcdir)/test.tex > $@
--- 56,63 ----
  lacheck.1:    lacheck.man
        -sed -e "s!%%LACHECKREV%%!Release $(REV)!" $(srcdir)/lacheck.man |\
        sed -e "s!%%LACHECKDATE%%!`date +%D`!" |\
!       sed -e "s!%%LACHECKPATH%%!$(bindir)/$(LACHECK)!" |\
!       nroff -man | gzip -9c > $@

  test.new:     test.tex $(LACHECK)
        $(srcdir)/$(LACHECK) $(srcdir)/test.tex > $@
***************
*** 68,73 ****
--- 69,77 ----
        -diff $(srcdir)/test.old $(srcdir)/test.new

  install:      $(LACHECK) lacheck.1
+       chown root.root lacheck lacheck.1
+       chmod 755 lacheck
+       chmod 644 lacheck.1
        cp $(srcdir)/$(LACHECK) $(bindir)
        cp $(srcdir)/lacheck.1 $(mandir)/lacheck$(manext)
```

56 LaTeX2html

Files:
*/usr/local/latex2html/**

Sources obtainable at:
`http://www-dsed.llnl.gov/files/programs/unix/latex2html/sources/latex2html-98.1p1.tar.gz`
`http://cbl.leeds.ac.uk/nikos/tex2html/doc/latex2html/latex2html.html`

The sources are Perl scripts, no compiling necessary. Untar from */usr/local*. Then change to the newly created directory and run `./install-test`. Accept the directory choice, and configure *pstoimg* as well. Specify `gif` as the image type, that is, type `g` when asked. Finally link *latex2html*

from */usr/local/bin*. Then link the files in *texinputs* from */usr/local/lib/texmf/tex/latex*, because that's where I keep local LaTeX input files.

The question arises, where to keep the icons that *latex2html* uses. As they are always the same icons, I decided to accept the default, which is to link translated documents to the icons in */usr/local/latex2html/icons.gif*. If one wants to get a local copy of the icons for a specific document, then run `latex2html` with the `-local_icons` flags.

I also chose to get mathematical equations set with the font recommended in the installation documentation, and I changed the scaling. For this it is necessary to edit *latex2html.config* and to set

```
$PK_GENERATION = 1;
$DVIPS_MODE = "toshiba";
$MATH_SCALE_FACTOR = 1.4;
$DISP_SCALE_FACTOR = 1.2;
$FIGURE_SCALE_FACTOR = 1.4;
```

this assumes that your TeX installation can deal with it.

You might need to run `install-test` if your Ghostscript binary or other support binaries from the *netpbm* package do not reside in */usr/local/bin*. Running it does not hurt even if they do.

Finally, I also installed a copy of the manual, which I downloaded separatedly. I wrote a short manpage, that points to this documentation.

The man page

```
.TH latex2html 1

.SH NAME

\fBlatex2html\fR \-\- LaTeX to HTML translator

.SH SYNOPSIS

.B latex2html [options] <file>.tex

.SH DESCRIPTION
.PP
\fBLatex2html\fR is a sophisticated LaTeX to HTML converter. It takes
as input a LaTeX file, and produces as output a subdirectory that
contains a HTML formatted version. More precisely, the input
\fBfile.tex\fR produces as output a directory \fBfile\fR, in which many
files will reside, and the top level document will be
\fBfile/file.html\fR.


.SH OPTIONS
The most important option is \fB-local_icons\fR, which tells the
converter to include a local copy of all its standard icons. The
default is that the produced HTML formatted documents refer to the
icons installed in /usr/local/latex2html/icons.gif. Documents pointing
to these icons can only be read from the local machine, and are not
```

```
suitable for posting on the web. For further options, read the HTML
documented documentation.

.SH DOCUMENTATION
LaTeX formatted documentation is found in /usr/local/latex2html/docs,
while HTML formatted documentation is found in
/usr/local/latex2html/manual. Read the HTML formatted documentation
with
.br
netscape file:/usr/local/latex2html/manual/manual.html &
.br
Of course, you can use a different browser instead of netscape.
```

57 Less

Files:
/usr/bin/{less,lessecho,lesskey}
/usr/man/man1/{less.1.gz,lesskey.1.gz}
/etc/profile

Sources obtainable at:
`http://ftp.gnu.ai.mit.edu/pub/gnu/less`

Compiles right out of the box. Can be installed directly (as shown below) or made into a Slackware package.

```
./configure --prefix=/usr
make
make install
strip /usr/bin/less /usr/bin/lessecho /usr/bin/lesskey
gzip -9 /usr/man/man1/less.1 /usr/man/man1/lesskey.1
```

Options that should always be valid can be set in an environment variable, as can be many other things. I also use *less* as the standard pager for the manual pages. I added the following to */etc/profile*.

```
LESS=-M
LESSCHARSET=latin1
PAGER='less -s'
export LESS LESSCHARSET PAGER
```

58 LessTif, a GNU-Motif for X

Files:
/usr/X11/bin/mxmkmf
/usr/X11/lib/X11/config/{Imake.tmpl, Motif.rules, Motif.tmpl}

/usr/X11/lib/{libXm., libMrm.*}*
*/usr/X11/LessTif/**
/usr/X11/include/{Xm, Mrm}

Sources obtainable at:
`http://www.lesstif.org/`
`http://sunsite.unc.edu/`

For a similar package see OpenMotif (section 82.2).

I built just the shared libraries. They will need updating, so there is no reason to link anything staticly. Also, I do not install any clients, only libraries, include files, and documentation. The default installation links version 1.2 as the default version for compilation. Total disk use of finished installation is about 5 MB. The documentation also includes documentation for the Xbae widget set, see Section 102.1, and the Xlt widget set. Finally make an entry into */usr/doc/html* for the documentation, and add the man pages to the MANPATH in */etc/profile*.

```
CFLAGS="-O2" ./configure --prefix=/usr/X11 --disable-debug
make -C include
make -C doc
make -C lib
# for self tests:
# make -C tests
make -C include install
make -C lib install
make -C doc install
gzip -9 /usr/X11/LessTif/doc/man/man?/*.?
ldconfig -v
# vi /etc/profile
# vi /usr/local/html/index.html
makewhatis -v /usr/X11/LessTif/doc/man
```

mxmkmf

mxmkmf is simply a shell script that first looks for the template for imake that contains the LessTif information (stored in */usr/X11/lib/X11/config*, and then for the standard template (which we don't have). It is buggy and must be fixed to read as follows.

```
#!/bin/sh
# LessTif version of xmkmf.
# $Id: mxmkmf.in,v 1.2 1997/02/18 07:07:46 u27113 Exp $
#
# Make sure to read the LessTif configuration first, to pick up
# that version of Imake.tmpl
#
imake -DUseInstalled -I/usr/X11/lib/X11/config
```

59 Libjpeg

Files:
/usr/local/lib/{libjpeg.a, libjpeg.so, libjpeg.so.6, libjpeg.so.6.a}

/usr/local/include/{jconfig.h, jmorecfg.h, jerror.h, jpeglib.h}
/usr/local/bin/{cjpeg, djpeg, jpegtran, rdjpgcom, wrjpgcom}
/usr/local/man/man1/{cjpeg.1.gz, jpeg.1.gz, djpeg.1.gz, jpegtran.1.gz, rdjpgcom.1.gz, wrjpgcom.1.gz}

Sources obtainable at:
`ftp://ftp.cs.columbia.edu/jpeg/src/jpegsrc.v6a.tar.gz`
`http://sunsite.unc.edu`
`ftp://ftp.x.org`

First of all, remove any old installation in */usr/bin*.

If you only want the static library do the following. Unpack the source, then run `./configure` and `make`. Install with `make install` and `make install-lib`, strip the binaries, and compress the manual pages.

If you want the static and the shared library, then do the following. Unpack the archive *jpegsrc.v6a.tar.gz* and cd into the newly created subdirectory, run `./configure`. Then edit the *Makefile*, so you create the shared library as well.

```
CFLAGS = -O3 -fomit-frame-pointer -fPIC -I$(srcdir)

all:  libjpeg.so.6 libjpeg.a cjpeg djpeg jpegtran rdjpgcom wrjpgcom

libjpeg.so.6: libjpeg.so.6a
        ln -sf libjpeg.so.6a libjpeg.so.6

libjpeg.so.6a:  $(LIBOBJECTS)
        $(CC) -shared -Wl,-soname,libjpeg.so.6 -o libjpeg.so.6a $(LIBOBJECTS)

cjpeg: $(COBJECTS) libjpeg.so.6
        $(LN) $(LDFLAGS) -o cjpeg $(COBJECTS) libjpeg.so.6 $(LDLIBS)

djpeg: $(DOBJECTS) libjpeg.so.6
        $(LN) $(LDFLAGS) -o djpeg $(DOBJECTS) libjpeg.so.6 $(LDLIBS)

jpegtran: $(TROBJECTS) libjpeg.so.6
        $(LN) $(LDFLAGS) -o jpegtran $(TROBJECTS) libjpeg.so.6 $(LDLIBS)
```

Run `make` and `make install; make install-lib`. You'll have to install the shared library by hand. Then run `ldconfig`, strip the installed binaries, and compress the man pages.

60 Libpng

Files:
/usr/local/lib/{ libpgn.a libpgn.so libpgn.so.0 libpgn.so.0.96 libpgn.so.2 libpgn.so.2.1.0.3 }
/usr/local/include/{ png.h pngconf.h }

Sources obtainable at:
`ftp://swrinde.nde.swri.edu/pub/png/src/libpng-0.96.tar.gz`

This library can only be compiled after *libz* (see Section 51.1.5) is installed.

Version 0.96

Unpack the archive *libpng-0.96.tar.gz* and cd into the newly created subdirectory, and edit the *makefile*:

```
CFLAGS= -O3 -fomit-frame-pointer -fPIC

all: libpng.so.0 libpng.a pngtest

libpng.so.0: $(OBJS)
        $(CC) -shared -Wl,-soname,libpng.so.0 -o libpng.so.0.96 $(OBJS)
        ln -sf libpng.so.0.96 libpng.so.0
```

and then run `make`, `make test`, and `make install`. I keep this version because some binaries are linked against it.

Version 1.03

Unpack the archive *libpng-1.0.3.tar.gz* and cd into the newly created subdirectory, and run `cp scripts/makefile.lnx makefile` and then edit the *makefile*:

```
# Where the zlib library and include files are located
ZLIBLIB=/usr/local/lib
ZLIBINC=/usr/local/include
#ZLIBLIB=../zlib
#ZLIBINC=../zlib

ALIGN=
# For I-386:
ALIGN=-malign-loops=2 -malign-functions=2
```

and then run `make`, `make test`, and `make install`.

61 Libtiff

Files:
/usr/local/lib/{*libtiff.a, libtiff.so, libtiff.so.3, libtiff.so.3.4.037*}
/usr/local/include/{*tiff.h, tiffio.h, tiffcomp.h, tiffiop.h, tiffconf.h*}

Sources obtainable at:
`ftp://ftp.sgi.com/graphics/tiff/tiff-v3.4*.gz`

Install *libjpeg* and *libz* first, see Sections 51.1.5 and 51.1.5. Unpack the archive *tiff-v3.4beta037-tar* and cd into the newly created subdirectory, and run `./configure`. Then cd into *libtiff*, run `chmod 644 Makefile`, and edit the Makefile to get the desired compiler flags.

```
CONF_LIBRARY= -DJPEG_SUPPORT -DZIP_SUPPORT
COPTS   =
OPTIMIZER=-O3 -fomit-frame-pointer -fPIC
```

Then run `make libtiff.a; make; make install`, afterwards run `ldconfig`.

62 Libtool

Files:
*/usr/local/libtool/**

Sources obtainable at:
`ftp://ftp.gnu.ai.mit.edu/pub/gnu/libtool`

This package is not needed to simply install GNU software, it is needed by the maintainers of GNU software. Hence I uninstalled it.

In the past, if a source code package developer wanted to take advantage of the power of shared libraries, he needed to write custom support code for each platform on which his package ran. He also had to design a configuration interface so that the package installer could choose what sort of libraries were built.

GNU Libtool simplifies the developer's job by encapsulating both the platform-specific dependencies, and the user interface, in a single script. GNU Libtool is designed so that the complete functionality of each host type is available via a generic interface, but nasty quirks are hidden from the programmer.

GNU Libtool's consistent interface is reassuring... users don't need to read obscure documentation in order to have their favorite source package build shared libraries. They just run your package 'configure' script (or equivalent), and libtool does all the dirty work.

```
CFLAGS="-O2" ./configure --prefix=/usr/local/libtool
make
make install
```

Now compress the info pages, link everything from */usr/local*, and make entries into the info directory file */usr/info/dir*.

63 Libz

Files:
/usr/local/lib/{*libz.a, libz.so, libz.so.1, libz.so.1.1.3*}
/usr/local/include/{*zconf.h, zlib.h*}

Sources obtainable at:
`http://www.cdrom.com/pub/infozip/zlib/`
`ftp://ftp.cdrom.com/pub/infozip/zlib/`
`ftp://swrinde.nde.swri.edu/pub/png/src/zlib-1.1.3.tar.gz`

Unpack the archive and cd into the newly created subdirectory. I prefer to make both the static and the dynamically loaded libraries.

```
# the static library first
./configure
make test
```

```
make install
# now the dynamically loaded library
./configure -s
make test
make install
```

64 LILO

Files:
/etc/lilo.conf

Lilo is the Linux loader. It installs a boot stanza on the first track of the hard drive. It needs to be run whenever its configuration file has changed or the kernel image has changed or physically moved on the disk, so in particular after recompiling the kernel and after repairing the hard drive from a floppy. If the hard drive is mounted on root (the directory /) then run `/sbin/lilo`. If the hard drive is mounted on */mnt*, then run `/mnt/sbin/lilo -r /mnt`. If you just want to know what lilo would do run `/sbin/lilo -t -v`.

Of course, lilo can also be used to install a boot stanza on a floppy. If the floppy has the necessary rudimentary file system, and has a valid configuration file in */etc/lilo.conf*, then mount the floppy on */mnt* and type `/sbin/lilo -r /mnt`.

Here is */etc/lilo.conf* from my hard drive.

```
# LILO configuration file
# generated by 'liloconfig'
#
# Start LILO global section
boot = /dev/hda2
#compact        # faster, but won't work on all systems.
delay = 50
# Normal VGA console
vga = normal
# ramdisk = 0     # paranoia setting
# End LILO global section
# Linux bootable partition config begins
image = /vmlinuz
  root = /dev/hda2
  label = linux
  read-only # Non-UMSDOS filesystems should be mounted read-only for checking
# Linux bootable partition config ends
# Linux bootable partition config begins
image = /boot/bzBackup
  root = /dev/hda2
  label = backup
  read-only # Non-UMSDOS filesystems should be mounted read-only for checking
# Linux bootable partition config ends
# Linux bootable partition config begins
image = /vmlinuz.old
  root = /dev/hda2
```

```
  label = old
  read-only # Non-UMSDOS filesystems should be mounted read-only for checking
# Linux bootable partition config ends
# DOS bootable partition config begins
other = /dev/hda1
  label = win98
  table = /dev/hda
# DOS bootable partition config ends
```

65 Luxman (PacMan for Linux)

Files:
/usr/games/{luxchk, luxman, luxman-snd}
*/usr/games/lib/luxman/**
/usr/man/man6/{luxman.6.gz, luxchk.6.gz}

The sources are from the Linux Slackware 3.2 distribution CD set. Unpack and run `make` and `make install`. Then compress the man pages. It needs the svga library installed, and only runs in a text console (non-X).

66 Lynx

Files:
*/usr/local/lib/lynx**
/usr/local/bin/lynx
/usr/local/man/man1/lynx.1.gz

Sources obtainable at:
`http://lynx.browser.org`

Needs configuration before compilation. The files *lynx.cnf* and *userdefs.h* need to be adjusted, see remarks below. Then

```
CFLAGS="-O2" ./configure --with-screen=ncurses --with-zlib\
  --prefix=/usr/local
make
make install
make install-help
```

The compress the man page, remove the old installation which is kept as **.old*-files, and fix all the permissions and ownerships.

The patch

```
diff -cr lynx2-8/lynx.cfg lynx2-8-local/lynx.cfg
*** lynx2-8/lynx.cfg Fri Mar  6 11:04:16 1998
--- lynx2-8-local/lynx.cfg Thu Oct 22 11:11:05 1998
***************
*** 22,28 ****
```

```
  #  replace PATH_TO with the complete path to FILENAME
  #  use Unix SHELL syntax and include the device on VMS systems)
  #
! STARTFILE:http://lynx.browser.org/

  # HELPFILE must be defined as a URL and must have a
  # complete path if local:
--- 22,29 ----
  #  replace PATH_TO with the complete path to FILENAME
  #  use Unix SHELL syntax and include the device on VMS systems)
  #
! # STARTFILE:http://lynx.browser.org/
! # STARTFILE:file://localhost/usr/local/lib/lynx_help/lynx_help_main.html

  # HELPFILE must be defined as a URL and must have a
  # complete path if local:
***************
*** 34,41 ****
  # http://www.crl.com/~subir/lynx/lynx_help/lynx_help_main.html
  #   This should be changed to the local path.
  #
! HELPFILE:http://www.crl.com/~subir/lynx/lynx_help/lynx_help_main.html
! #HELPFILE:file://localhost/PATH_TO/lynx_help/lynx_help_main.html

  # DEFAULT_INDEX_FILE is the default file retrieved when the
  # user presses the 'I' key when viewing any document.
--- 35,43 ----
  # http://www.crl.com/~subir/lynx/lynx_help/lynx_help_main.html
  #   This should be changed to the local path.
  #
! # HELPFILE:http://www.crl.com/~subir/lynx/lynx_help/lynx_help_main.html
! # HELPFILE:file://localhost/PATH_TO/lynx_help/lynx_help_main.html
! # HELPFILE:file://localhost/usr/local/lib/lynx_help/lynx_help_main.html

  # DEFAULT_INDEX_FILE is the default file retrieved when the
  # user presses the 'I' key when viewing any document.
***************
*** 111,117 ****
  # This definition will be overridden if a "LYNX_SAVE_SPACE" environment
  # variable has been set on Unix, or logical has been defined on VMS.
  #
! #SAVE_SPACE:~/foo/

  # If LYNX_HOST_NAME is defined here or in userdefs.h, it will be
  # treated as an alias for the local host name in checks for URLs on
--- 113,119 ----
  # This definition will be overridden if a "LYNX_SAVE_SPACE" environment
  # variable has been set on Unix, or logical has been defined on VMS.
  #
! #SAVE_SPACE:""

  # If LYNX_HOST_NAME is defined here or in userdefs.h, it will be
```

```
  # treated as an alias for the local host name in checks for URLs on
***************
*** 120,126 ****
  # name of the system on which Lynx is running) will all be passed as
  # local.  A different definition here will override that in userdefs.h.
  #
! #LYNX_HOST_NAME:www.cc.ukans.edu

  # localhost aliases
  # Any LOCALHOST_ALIAS definitions also will be accepted as local when
--- 122,128 ----
  # name of the system on which Lynx is running) will all be passed as
  # local.  A different definition here will override that in userdefs.h.
  #
! #LYNX_HOST_NAME:localhost

  # localhost aliases
  # Any LOCALHOST_ALIAS definitions also will be accepted as local when
***************
*** 139,145 ****
  # if your system does not have utmp capabilities.  CHANGE THIS here
  # if it was not changed in userdefs.h at compilation time.
  #
! #LOCAL_DOMAIN:ukans.edu

  # CHARACTER_SET defines the default character set, i.e., that assumed
  # to be installed on the user's terminal.  It determines which characters
--- 141,147 ----
  # if your system does not have utmp capabilities.  CHANGE THIS here
  # if it was not changed in userdefs.h at compilation time.
  #
! #LOCAL_DOMAIN:vanderbilt.edu

  # CHARACTER_SET defines the default character set, i.e., that assumed
  # to be installed on the user's terminal.  It determines which characters
***************
*** 342,348 ****
  # are defined in userdefs.h, and can be modified here should longer
  # pauses be desired for braille-based access to Lynx.
  #
! #INFOSECS:1
  #MESSAGESECS:2
  #ALERTSECS:3

--- 344,350 ----
  # are defined in userdefs.h, and can be modified here should longer
  # pauses be desired for braille-based access to Lynx.
  #
! #INFOSECS:2
  #MESSAGESECS:2
  #ALERTSECS:3
```

```
***************
*** 704,710 ****
  # (e.g., ./lynx/.lynxsig).  The definition is set in userdefs.h and can
  # be changed here.
  #
! #LYNX_SIG_FILE:.lynxsig

  # If USE_MOUSE is set TRUE, Lynx (when configured with ncurses) will allow
  # the user to click with button-1 on links to select them.
--- 706,712 ----
  # (e.g., ./lynx/.lynxsig).  The definition is set in userdefs.h and can
  # be changed here.
  #
! #LYNX_SIG_FILE:.signature

  # If USE_MOUSE is set TRUE, Lynx (when configured with ncurses) will allow
  # the user to click with button-1 on links to select them.
diff -cr lynx2-8/userdefs.h lynx2-8-local/userdefs.h
*** lynx2-8/userdefs.h Tue Mar 10 06:07:58 1998
--- lynx2-8-local/userdefs.h Thu Oct 22 11:10:52 1998
***************
*** 280,286 ****
   * These global and personal files override anything in
   * lynx.cfg or src/HTInit.c
   */
! #define GLOBAL_EXTENSION_MAP "/usr/local/lib/mosaic/mime.types"
  #define PERSONAL_EXTENSION_MAP ".mime.types"

  /*************************
--- 280,286 ----
   * These global and personal files override anything in
   * lynx.cfg or src/HTInit.c
   */
! #define GLOBAL_EXTENSION_MAP "/usr/local/lib/mime.types"
  #define PERSONAL_EXTENSION_MAP ".mime.types"

  /*************************
***************
*** 289,295 ****
   * These global and personal files override anything in
   * lynx.cfg or src/HTInit.c
   */
! #define GLOBAL_MAILCAP "/usr/local/lib/mosaic/mailcap"
  #define PERSONAL_MAILCAP ".mailcap"

  /*************************
--- 289,295 ----
   * These global and personal files override anything in
   * lynx.cfg or src/HTInit.c
   */
! #define GLOBAL_MAILCAP "/usr/local/lib/mailcap"
  #define PERSONAL_MAILCAP ".mailcap"
```

```
  /**************************
***************
*** 322,328 ****
   * open is used as the default for NeXT, instead of the XLOADIMAGE_COMMAND
   * definition.
   */
! #define XLOADIMAGE_COMMAND "xli %s &"

  /**************************
   * For UNIX systems this should be sendmail
--- 322,328 ----
   * open is used as the default for NeXT, instead of the XLOADIMAGE_COMMAND
   * definition.
   */
! #define XLOADIMAGE_COMMAND "xv %s &"

  /**************************
   * For UNIX systems this should be sendmail
***************
*** 416,422 ****
   *  configuration default can be toggled via the -core command
   *  line switch.
   */
! #define NO_FORCED_CORE_DUMP FALSE

  /**************************
   * LYNX_LSS_FILE is the location and name of the default lynx
--- 416,422 ----
   *  configuration default can be toggled via the -core command
   *  line switch.
   */
! #define NO_FORCED_CORE_DUMP TRUE

  /**************************
   * LYNX_LSS_FILE is the location and name of the default lynx
***************
*** 450,456 ****
   * note: STARTFILE must be a URL.  See the Lynx online help for more
   *       information on URLs
   */
! #define STARTFILE "http://lynx.browser.org/"

  /****************************
   * HELPFILE must be defined as a URL and must have a
--- 450,457 ----
   * note: STARTFILE must be a URL.  See the Lynx online help for more
   *       information on URLs
   */
! /* #define STARTFILE "http://lynx.browser.org/" */
! #define STARTFILE "file://localhost/usr/local/lib/lynx_help/lynx_help_main.htm
l"
```

```
  /*****************************
   * HELPFILE must be defined as a URL and must have a
***************
*** 463,470 ****
   * http://www.crl.com/~subir/lynx/lynx_help/lynx_help_main.html
   *   This should be changed here or in lynx.cfg to the local path.
   */
! #define HELPFILE "http://www.crl.com/~subir/lynx/lynx_help/lynx_help_main.html
"
! /* #define HELPFILE "file://localhost/PATH_TO/lynx_help/lynx_help_main.html" *
/

  /*****************************
   * DEFAULT_INDEX_FILE is the default file retrieved when the
--- 464,471 ----
   * http://www.crl.com/~subir/lynx/lynx_help/lynx_help_main.html
   *   This should be changed here or in lynx.cfg to the local path.
   */
! /* #define HELPFILE "http://www.crl.com/~subir/lynx/lynx_help/lynx_help_main.h
tml" */
! #define HELPFILE "file://localhost/usr/local/lib/lynx_help/lynx_help_main.html
"

  /*****************************
   * DEFAULT_INDEX_FILE is the default file retrieved when the
***************
*** 572,578 ****
   * if your system does not have utmp capabilities.  CHANGE THIS here
   * or in lynx.cfg.
   */
! #define LOCAL_DOMAIN "ukans.edu"

  /*******************************
  * The DEFAULT_CACHE_SIZE specifies the number of WWW documents to be
--- 573,579 ----
   * if your system does not have utmp capabilities.  CHANGE THIS here
   * or in lynx.cfg.
   */
! #define LOCAL_DOMAIN "vanderbilt.edu"

  /*******************************
  * The DEFAULT_CACHE_SIZE specifies the number of WWW documents to be
***************
*** 599,605 ****
  * number of documents to cache (rather than the maximum number only if
  * DEFAULT_VIRTUAL_MEMORY_SIZE has been exceeded, as with VAXC/VAX).
  */
! #define DEFAULT_CACHE_SIZE 10

  #if defined(VMS) && defined(VAXC) && !defined(__DECC)
  #define DEFAULT_VIRTUAL_MEMORY_SIZE 512000
```

```
--- 600,606 ----
  * number of documents to cache (rather than the maximum number only if
  * DEFAULT_VIRTUAL_MEMORY_SIZE has been exceeded, as with VAXC/VAX).
  */
! #define DEFAULT_CACHE_SIZE 20

  #if defined(VMS) && defined(VAXC) && !defined(__DECC)
  #define DEFAULT_VIRTUAL_MEMORY_SIZE 512000
***************
*** 807,813 ****
   * directory.  If it is in a subdirectory, begin it with a dot-slash
   * (e.g., ./lynx/.lynxsig).  The definition here can be changed in lynx.cfg.
   */
! #define LYNX_SIG_FILE ".lynxsig"

  /*******************************
   * If USE_SELECT_POPUPS is set FALSE, Lynx will present a vertical list
--- 808,814 ----
   * directory.  If it is in a subdirectory, begin it with a dot-slash
   * (e.g., ./lynx/.lynxsig).  The definition here can be changed in lynx.cfg.
   */
! #define LYNX_SIG_FILE ".signature"

  /*******************************
   * If USE_SELECT_POPUPS is set FALSE, Lynx will present a vertical list
***************
*** 861,867 ****
   * defined here can be modified via lynx.cfg, should longer pauses be
   * desired for braille-based access to Lynx.
   */
! #define INFOSECS 1
  #define MESSAGESECS 2
  #define ALERTSECS 3

--- 862,868 ----
   * defined here can be modified via lynx.cfg, should longer pauses be
   * desired for braille-based access to Lynx.
   */
! #define INFOSECS 2
  #define MESSAGESECS 2
  #define ALERTSECS 3
```

67 LyX

Files:
/usr/local/bin/lyx
/usr/local/man/man1/lyx.1.gz
*/usr/local/lib/lyx/**

Sources obtainable at:

`http://www-pu.informatik.uni-tuebingen.de/users/ettrich/lyx.html`
`http://la1ad.uio.no/~larsbj/lyx.shtml`

Compile with `./configure` and `make all CXXFLAGS='-O2 -m486'` and then install with `make install`. Compress the man page, and you're done.

How do I tell LyX that my LaTeX configuration has changed?

Run `latex chkconfig.ltx` from the LyX system dir, per default */usr/local/lib/lyx*. This will inspect your LaTeX installation and update the system as needed. You can inspect the result of inspection by reading the generated file *LaTeXConfig.lyx* which you'll find in the Help/Documentation browser.

68 Mac Utilities

Files:
/usr/local/bin/{*xbin, mcvert, macunpack*}
/usr/local/man/man1/{*xbin.1.gz, mcvert.1.gz, macunpack.1.gz*}

Sources obtainable at:
search the Internet for `binhex`, that's how `xbin` is known
you find `mcvert` and `macunpack` in any CTAN archive under /pub/tex/tools

Usually *xbin* is found as an email message including the source code, just remove the email header and footer, and compile. Then write a little man page.

For *mcvert* and *macunpack* get the corresponding subdirectory from a CTAN mirror. Then edit the makefiles to get the compiler flags you want, and then make. For *macunpack* you need to do this from the *unpack* subdirectory. Install man pages and binaries by hand. I chose to rename the *unpack* binary to *macunpack*.

69 Make

Files:
/usr/bin/make
/usr/man/man1/make.1.gz
*/usr/info/make.info**

Sources obtainable at:
`ftp://ftp/gnu.ai.mit.edu/pub/gnu/make/`

make

The new version of `make` compiles straight out of the box.

```
./configure --prefix=/usr --disable-nls
make CFLAGS="-O2 -s"
```

```
make check
mv /usr/bin/make /usr/bin/make.old
make.old install
rm /usr/bin/make.old
gzip -9 /usr/info/make.info* /usr/man/man1/make.1
```

mk

I use certain commands to clean up all the time and collected them into a make file. Important: the lines in the make file have to start with a `Tab`, and not with spaces. The minus signs cause any errors (like file not found) to be ignored. I have an `alias mk='make -f ~/.Makefile'` so I can use `mk clean` from any directory. So here is that file *${HOME}/.Makefile*.

```
RMF = rm -f
RMI = rm -i

nothing:

clean:
-$(RMF) *.log*
-$(RMF) *.spell
-$(RMF) *~
-$(RMF) .*~
-$(RMF) \#*
-$(RMF) core
-$(RMF) *.aux
-$(RMF) *.run

clobber: clean
-$(RMF) *.lof
-$(RMF) *.lot
-$(RMF) *.toc
-$(RMF) *.bbl
-$(RMF) *.blg
-$(RMI) *.dvi*
-$(RMI) *.ps
-$(RMI) *.ps.gz
-$(RMI) *.pdf
-$(RMI) *.bak
-$(RMI) *.g3
```

70 Manual Pages

Files:
*/usr/man/**
*/usr/local/man/**
*/usr/man/nonlinux/**
/usr/man/whatis

There is a package containing the Linux man pages for all chapters except 1 and 8. The package is called man1.10 or so, and is available from your nearest Linux ftp site. Install by copying to your favorite location. `make install` will copy them to */usr/man/man[1-9]*. If your man pager can handle compressed source pages (mine can), you might do `make gz` before `make install`.

Manual pages that come with software are installed into */usr/local/man*. Manual pages copied from other systems go into */usr/man/nonlinux*. I have a few man pages from BSD and a few from HP.

The environmental variable `MANPATH` givens the location of all man pages. It is set at login in */etc/profile*. The path is searched from left to right, and if you have two versions of a man page with the same topic, you can override the path by using the `-M` flag when using `man`. For example, `man vi` gets me the man page that came with Linux, while `man -M /usr/man/nonlinux vi` gets me a more detailed man page which I copied from an HP system.

The file */usr/man/whatis* contains a database suitable for searching with `apropos`. To update the database do the following.

```
mv /usr/man/whatis /usr/man/whatis.old # safety measure
/usr/lib/makewhatis -c -w        # uses the cat pages and the current man path
mv /usr/man/whatis /usr/man/whatis.cat
/usr/lib/makewhatis -w           # uses the man pages and the current man path
mv /usr/man/whatis /usr/man/whatis.man
cd /usr/man
cat whatis.old whatis.cat whatis.man | sort | uniq > /usr/man/whatis
```

Warning: There is a *whatis* file in each man directory.

71 Mpack

Files:
/usr/bin/{mpack munpack}
/usr/man/man1/{mpack.1.gz munpack.1.gz}
*/usr/doc/mpack-1.5/**

Sources obtainable at:
`http://linux01.gwdg.de/pub/mpack/`

Edit the *Makefile* to set `OPT=-O2` and tt DESTDIR=/usr. Compile and install as usual with `make` and `make install`. The strip binaries, compress the man pages, and install the file *README.unix* into */usr/local/doc/mpack-1.5*. I made mpack into a Slackware package and intalled with *pkgtool* (see Section 102.1).

72 Mgetty and Sendfax

Files:
*/usr/local/mgetty+sendfax/**

Sources obtainable at:
`http://www.leo.org/~doering/mgetty/`
`http://sunsite.unce.edu/pub/Linux/system/serial/getty`

Untar the source, `cd` to the created directory, and `cp policy.h-dist policy.h; chmod 644 Makefile`. Then edit those two files, see changes below. After editing, run `make` and `make install`. I also chose to install *viewfax*, which is in *frontends/X11* subdirectory. Compiles as stated in the README, that is with the sequence:

```
cd frontends/X11/viewfax-2.4
xmkmf
make depend
make
make install
make install.man
```

However, I edited the *Imakefile* to get *viewfax* installed into the correct location, see changes below.

In the binaries *faxq, faxrunq, faxspool,* and *faxrm* one needs to set `echo` to some program that understands escapes, so either `/bin/echo` or `echo -e`. If the changes to the Makefile were done as indicated below, then this is done automatically and does not need to be done by hand.

Compress the man and info pages, and make an entry to the info directory file */usr/info/dir*. Finally, add a line to */etc/inittab* if you want to be able to dial in to your computer. Then I added the file *fax.allow* with the entries `root`, `mayer`, and `souders` to the *etc/mgetty+sendfax* directory. Note that faxes are only send when *faxrunq* is run, which is only run-able by `root`. One could put a `crontab` entry that runs `faxrunq` every once in a while.

As always, I installed the whole distribution into its own subdirectory, at the end link everything from */usr/local*.

Finally, I went to the homepage of mgetty+sendfax and downloaded the manual in HTML format. I compressed the pages after fixing all cross-references, and installed it into this directory as well. I then linked it from the index in */usr/local/html*. The commands are given below.

```
cd /usr/local/mgetty+sendfax
mkdir html
cd html
tar -zxvf manual.html.tar.gz
for i in *.html; do
  sed -e 's/"mgetty_toc.html/"mgetty_toc.html.gz/g' \
      -e 's/"mgetty_\([0-9]*\).html/"mgetty_\1.html.gz/g' $i > foo \
  && cat foo > $i
done
rm -f foo
gzip -9 *
```

The line added to /etc/inittab

```
# d3:5:respawn:/usr/local/sbin/mgetty -x 2 ttyS1
```

Changes to policy.h

```
*** policy.h Sun Feb 28 21:54:26 1999
--- policy.h-good Sun Nov 21 20:45:59 1999
***************
*** 96,102 ****
```

```
  /* group id that the device is chown()ed to. If not defined, the
   * primary group of "DEVICE_OWNER" is used.
   */
! #define DEVICE_GROUP "modem"

  /* access mode for the line while getty has it - it should be accessible
   * by uucp / uucp, but not by others (imagine someone dialing into your
--- 96,102 ----
  /* group id that the device is chown()ed to. If not defined, the
   * primary group of "DEVICE_OWNER" is used.
   */
! #define DEVICE_GROUP "uucp"

  /* access mode for the line while getty has it - it should be accessible
   * by uucp / uucp, but not by others (imagine someone dialing into your
***************
*** 225,231 ****
   *
   * Depending on your system, "/var/run/mgetty.%s" might be a good place.
   */
! #define MGETTY_PID_FILE "/etc/mg-pid.%s"

  /* Path for the lock files. A %s will be replaced with the device name,
   * e.g. tty2a -> /usr/spool/uucp/LCK..tty2a
--- 225,231 ----
   *
   * Depending on your system, "/var/run/mgetty.%s" might be a good place.
   */
! #define MGETTY_PID_FILE "/var/run/mgetty.%s"

  /* Path for the lock files. A %s will be replaced with the device name,
   * e.g. tty2a -> /usr/spool/uucp/LCK..tty2a
***************
*** 418,424 ****
   * in most of the received faxes.
   * Most faxmodems expect Xon/Xoff, few honour the RTS line.
   */
! #define FAXREC_FLOW FLOW_HARD | FLOW_SOFT

  /* And this is for sending faxes
   *
--- 418,424 ----
   * in most of the received faxes.
   * Most faxmodems expect Xon/Xoff, few honour the RTS line.
   */
! #define FAXREC_FLOW FLOW_SOFT

  /* And this is for sending faxes
   *
***************
*** 437,443 ****
   * that fax sending will time out after the first page sent (no ACK received)
```

```
   * and fail if FLOW_HARD is used. Use FLOW_SOFT instead.
   */
! #define FAXSEND_FLOW FLOW_HARD | FLOW_SOFT

  /* if your faxmodem switches port bit rate just after sending the "+FCON"
   * message to the host, define this to contain the baudrate used. (Not
--- 437,443 ----
   * that fax sending will time out after the first page sent (no ACK received)
   * and fail if FLOW_HARD is used. Use FLOW_SOFT instead.
   */
! #define FAXSEND_FLOW FLOW_SOFT

  /* if your faxmodem switches port bit rate just after sending the "+FCON"
   * message to the host, define this to contain the baudrate used. (Not
***************
*** 472,478 ****
   * but some do only allow digits and blank
   * AT+FLID=? should tell you what's allowed and what not.
   */
! #define FAX_STATION_ID "49 115 xxxxxxxx"

  /* ------ sendfax-specific stuff follows here -------- */

--- 472,478 ----
   * but some do only allow digits and blank
   * AT+FLID=? should tell you what's allowed and what not.
   */
! #define FAX_STATION_ID "YOUR NUMBER HERE"

  /* ------ sendfax-specific stuff follows here -------- */

***************
*** 512,518 ****
   * need something like "ATx0DT0wP" (switch of dial-tone recognition, tone-
   * dial a "0", wait for dial-tone, pulse dial the rest)
   */
! #define FAX_DIAL_PREFIX "ATD"

  /* When sending a fax, if the other side says "page bad, retrain
   * requested", sendfax will retry the page. Specifiy here the maximum
--- 512,518 ----
   * need something like "ATx0DT0wP" (switch of dial-tone recognition, tone-
   * dial a "0", wait for dial-tone, pulse dial the rest)
   */
! #define FAX_DIAL_PREFIX "ATDT"

  /* When sending a fax, if the other side says "page bad, retrain
   * requested", sendfax will retry the page. Specifiy here the maximum
***************
*** 534,540 ****
   * If you don't adapt this for your needs, sendfax won't run (you can
   * set it from the sendfax.config file, though)!
```

```
   */
! #define FAX_MODEM_TTYS "tty4c:tty4d"

  /* Xon or not?
   *
--- 534,540 ----
   * If you don't adapt this for your needs, sendfax won't run (you can
   * set it from the sendfax.config file, though)!
   */
! #define FAX_MODEM_TTYS "ttyS1"

  /* Xon or not?
   *
***************
*** 577,583 ****
  /* where to send notify mail about incoming faxes to
   * (remember to create an mail alias if no such user exists!)
   */
! #define MAIL_TO "faxadmin"

  /* after a fax has arrived, mgetty can call a program for further
   * processing of this fax.
--- 577,583 ----
  /* where to send notify mail about incoming faxes to
   * (remember to create an mail alias if no such user exists!)
   */
! #define MAIL_TO "root"

  /* after a fax has arrived, mgetty can call a program for further
   * processing of this fax.
***************
*** 592,598 ****
   * If you don't want this type of service, do not define it at all
   * Absolute path name has to be used here!
   */
! #define FAX_NOTIFY_PROGRAM "/usr/local/lib/mgetty+sendfax/new_fax"

  /* default minimum space required on spooling partition for receiving a FAX
   * (in KILObytes)
--- 592,598 ----
   * If you don't want this type of service, do not define it at all
   * Absolute path name has to be used here!
   */
! #define FAX_NOTIFY_PROGRAM "/usr/local/mgetty+sendfax/bin/new_fax"

  /* default minimum space required on spooling partition for receiving a FAX
   * (in KILObytes)
```

Changes to Makefile

```
*** Makefile Sat Jul 24 23:28:18 1999
--- Makefile-good Sun Nov 21 20:13:50 1999
```

```
***************
*** 177,183 ****
  #
  # prefix, where most (all?) of the stuff lives, usually /usr/local or /usr
  #
! prefix=/usr/local
  #
  # prefix for all the spool directories (usually /usr/spool or /var/spool)
  #
--- 177,183 ----
  #
  # prefix, where most (all?) of the stuff lives, usually /usr/local or /usr
  #
! prefix=/usr/local/mgetty+sendfax
  #
  # prefix for all the spool directories (usually /usr/spool or /var/spool)
  #
***************
*** 247,253 ****
  # If you have Perl with TK extentions, define it here. This may be the
  # same as PERL=... above, or different, if you have TkPerl statically
  # linked.
! TKPERL=/usr/bin/tkperl
  #
  #
  # An echo program that understands escapes like "\n" for newline or
--- 247,253 ----
  # If you have Perl with TK extentions, define it here. This may be the
  # same as PERL=... above, or different, if you have TkPerl statically
  # linked.
! # TKPERL=/usr/bin/tkperl
  #
  #
  # An echo program that understands escapes like "\n" for newline or
***************
*** 259,265 ****
  # please use the "mg.echo" program provided in the compat/ subdirectory.
  # Set ECHO="mg.echo" and INSTALL_MECHO to mg.echo
  #
! ECHO="echo"
  #
  # INSTALL_MECHO=mg.echo

--- 259,265 ----
  # please use the "mg.echo" program provided in the compat/ subdirectory.
  # Set ECHO="mg.echo" and INSTALL_MECHO to mg.echo
  #
! ECHO="echo -e"
  #
  # INSTALL_MECHO=mg.echo
```

Changes to Imakefile of viewfax

```
*** frontends/X11/viewfax-2.4/Imakefile Mon Oct 16 20:18:38 1995
--- frontends/X11/viewfax-2.4/Imakefile-good Sun Nov 21 20:54:52 1999
***************
*** 5,13 ****
     complain about this.  Sheesh! :-) */
  HELPDIR = ${USRLIBDIR}/X11
  #else
! BINDIR = /usr/local/bin /* where you want the binary installed */
! MANDIR = /usr/local/man/man1 /* where you want the man file installed */
! HELPDIR = /usr/local/lib /* where you want the help file installed */
  #endif

  DEFINES = -DHELPFILE=\"$(HELPDIR)/viewfax.tif\"
--- 5,13 ----
     complain about this.  Sheesh! :-) */
  HELPDIR = ${USRLIBDIR}/X11
  #else
! BINDIR = /usr/local/mgetty+sendfax/bin /* where you want the
binary installed */
! MANDIR = /usr/local/mgetty+sendfax/man/man1 /* where you want the man file
 installed */
! HELPDIR = /usr/local/mgetty+sendfax/lib /* where you want the help fil
e installed */
  #endif

  DEFINES = -DHELPFILE=\"$(HELPDIR)/viewfax.tif\"
```

Changes to faxheader

Apply this patch after installation, not before.

```
*** faxheader Sun Nov 21 20:47:24 1999
--- faxheader-good Sun Nov 21 21:25:08 1999
***************
*** 1,2 ****

!       FAX  FROM:  **not configured** YOUR NUMBER HERE     TO: @T@    PAGE: @P
@ OF @M@
--- 1,3 ----

!
!   FROM: @N@  YOUR NUMBER HERE  @DATE@  TO: @T@  (@P@ OF @M@)
```

etc/mgetty+sendfax/fax.allow

```
root
TrustedUser1
TrustedUser2
```

73 Midnight Commander

Files:
/usr/bin/{mcserv, mcedit, mc, mcmfmt}
*/usr/lib/mc/**
/usr/man/man1/{mc.1.gz, mcedit.1.gz}
/usr/man/man8/mcserv.8.gz

The sources can be found on the GNU and on the linux archives.

```
CFLAGS="-O2" CC="gcc" ./configure --prefix=/usr --without-gpm-mouse
make
make install
```

Compress the man pages, strip the binaries, and get rid of all the files that get written into */usr/share/icons* and */usr/share/locale*.

I then installed the Slackware package. But it is based on a Gnome distribution, which expects Samba, which I do not have installed. So I recompiled. Here is the script.

The modified mc.build-lite

```
#!/bin/sh
CWD=‘pwd‘
cd /tmp
tar xzvf $CWD/mc-4.5.50.tar.gz
cd mc-4.5.50
( cd vfs/samba ; zcat $CWD/samba.codepages.diff.gz | patch -p0 )
CFLAGS=-O2 ./configure --prefix=/usr \
    --with-catgets \
    --disable-samba \
    --disable-nls \
            --disable-gpm-mouse \
            --with-ncurses=/usr \
    --with-slang \
            --with-ext2undel \
            --localstatedir=/var \
            --with-x=no \
            i386-slackware-linux
make
make install
mkdir -p /usr/doc/mc-4.5.50
cp -a FAQ COPYING NEWS README /usr/doc/mc-4.5.50
chown -R root.root /usr/doc/mc-4.5.50
```

74 MPlayer

Files:
/usr/local/bin/mplayer_save
*/usr/local/mplayer/**

Sources obtainable at:
`http://mplayer.dev.hu/homepage/`

I used the cvs instructions. Also, get the zip file with the windows codecs and unzip them into */usr/local/mplayer/win32lib*.

```
mkdir -p /usr/local/mplayer/bin /usr/local/mplayer/man/man1
mkdir -p /usr/local/mplayer/share/mplayer
./configure --prefix=/usr/local/mplayer \
--with-win32libdir=/usr/local/mplayer/win32lib
make
# Compilation gets stuck in final linking stages. Copy the command with the
# mouse and add /usr/lib/libstdc++-* at the end. This will do it.
make
cp DOCS/codecs.conf /usr/local/mplayer/share/mplayer/
make install
cd /usr/local/mplayer
rm bin/fibmap_mplayer                    # only for encrypted DVDs
gzip -9 /usr/local/man/man1/mplayer.1
```

Now link the whole thing from */usr/local* as always. I also installed a little script into */usr/local/bin* to save viewed videos on the fly.

mplayer_save

```
#!/bin/sh
save $1
mplayer -nosound > /dev/null $1
```

75 Modules, Patches, and Kernel Compiling

Files:
*/usr/src/linux/src/**

Applying patches

Patches must be applied from specific directories. For the kernel patches this is */usr/src*. That doesn't mean the patch files have to actually be there. So assume you have linux version 2.0.30 and you want to upgrade.

```
cd /usr/src
gzip -dc linux/src/patch-2.0.31.gz | patch -p0
mv linux-2.0.30 linux-2.0.31
ln -sf linux-2.0.31 linux
```

The `-p0` flag tells `patch` not to strip any path information stored in the patch file. The patches must be applied in their numeric order.
To double-check that all patches were applied correctly run

```
find /usr/src/linux -follow -name "*.rej" -print
find /usr/src/linux -follow -name "*#" -print
```

This will list any rejected portions of the patch process.

modules (old information)

A *module* is a piece of kernel code that is loaded when needed. `modules-2.0.0.tar.gz` is the source. I keep a copy in */usr/src/kernel/*. Unpack the modules from */usr/src*. Run `make all; make install`. This will copy uncompressed man pages to */usr/man*, compress them there if you wish. The man pages are *insmod.1, ksyms.1, lsmod.1, rmmod.1, modules.2, genksyms.8* and *depmod.1*. Messages of modules will appear in */var/adm/messages*. You can check if the modules work by running `make drv_hello.o` from the subdirectory *insmod*, and then running

```
insmod drv_hello.o
lsmod
rmmod drv_hello
```

This installs the module, lists all loaded modules, and removes it again. Also check for messages in */var/adm/messages*. The kernel will load any needed modules automatically, at least if it is configured correctly.

Then you can remove the directory */usr/src/modules-2.0.0*.

Module configuration

The file */etc/modules.conf* tells the kernel daemon *kerneld* in which order and for what to load modules. Instead of running *kerneld* it is possible to run a kernel thread called *kmod*. For this enable the `Kernel module loader` under the section `Loadable module support` in the kernel configuration. However, this thread does not unload modules, it only loads them. Hence it is necessary to have a *crontab* entry that periodically removes any unused modules. This is one such entry, it removes unused modules every quarter hour.

```
*/15 * * * * /sbin/rmmod -a >& /dev/null
```

Here is the file */etc/modules.conf*.

```
alias block-major-8 sd_mod
post-install ppp /sbin/modprobe "-k" "ppp_deflate"
pre-install ppa /sbin/modprobe "-k" "parport_pc"
pre-install lp /sbin/modprobe "-k" "parport_pc"
post-install sd_mod /sbin/modprobe "-k" "ppa"

# New soundconfiguration using Yamaha FM Synthesizer. Works, but not well.
alias char-major-14 opl3sa2
pre-install opl3sa2 modprobe "-k" "ad1848"
#post-install opl3sa2 modprobe "-k" "opl3"
options opl3 io=0x388
options opl3sa2 io=0x370 mss_io=0x530 irq=5 dma=1 dma2=0 mpu_io=0x330
```

```
# Old 8-bit soundblaster configuration. Works, but not well.
#
# alias char-major-14 sb
# post-install sb /sbin/modprobe "-k" "adlib_card"
# options sb io=0x220 irq=5 dma=0 dma16=0 mpu_io=0x330
# options mpu401 irq=5 io=0x330
# options adlib_card io=0x388     # FM synthesizer
# options opl3sa2 io=0x388 mss_io=0x530 irq=5 dma=1 dma2=0

keep
path[pcmcia]=/lib/modules/‘uname -r‘
path[pcmcia]=/lib/modules/preferred-‘uname -r‘
path[pcmcia]=/lib/modules/default
path[pcmcia]=/lib/modules/preferred
```

Building the kernel

This is very memory intensive. Always compile long programs without having X windows running, at least if you don't have more than, say, 16 MB of RAM. I currently have 96 MB, so I don't really care.

```
cd /usr/src/linux
make clean
make config
# or do "make xconfig" if you have tcl/tk
make dep
# if your boot script is set up to have the kernel in /vmlinuz, then
make zlilo
# otherwise install by hand:
# make zImage
# mv /boot/zImage /boot/zImage.old
# mv /usr/src/linux/arch/i386/boot/zImage /boot/zImage
# /sbin/lilo -v
make modules
make modules_install
```

After upgrading the kernel, it is also necesssary to upgrade/re-compile anything that creates kernel modules, in particular the PCMCIA package (Section 82.2).

Here is a listing of the configuration I have, stored in the file *.config* in the source directory. Read Section 102.1 for the configuration of the sound card. Read it before you start making the new kernel.

```
#
# Automatically generated make config: don't edit
#

#
# Code maturity level options
#
# CONFIG_EXPERIMENTAL is not set
```

```
#
# Processor type and features
#
# CONFIG_M386 is not set
# CONFIG_M486 is not set
# CONFIG_M586 is not set
# CONFIG_M586TSC is not set
CONFIG_M686=y
CONFIG_X86_WP_WORKS_OK=y
CONFIG_X86_INVLPG=y
CONFIG_X86_BSWAP=y
CONFIG_X86_POPAD_OK=y
CONFIG_X86_TSC=y
CONFIG_X86_GOOD_APIC=y
CONFIG_1GB=y
# CONFIG_2GB is not set
# CONFIG_MATH_EMULATION is not set
CONFIG_MTRR=y
# CONFIG_SMP is not set

#
# Loadable module support
#
CONFIG_MODULES=y
CONFIG_MODVERSIONS=y
CONFIG_KMOD=y

#
# General setup
#
CONFIG_NET=y
CONFIG_PCI=y
# CONFIG_PCI_GOBIOS is not set
# CONFIG_PCI_GODIRECT is not set
CONFIG_PCI_GOANY=y
CONFIG_PCI_BIOS=y
CONFIG_PCI_DIRECT=y
CONFIG_PCI_QUIRKS=y
# CONFIG_PCI_OLD_PROC is not set
# CONFIG_MCA is not set
# CONFIG_VISWS is not set
CONFIG_SYSVIPC=y
# CONFIG_BSD_PROCESS_ACCT is not set
CONFIG_SYSCTL=y
CONFIG_BINFMT_AOUT=y
CONFIG_BINFMT_ELF=y
# CONFIG_BINFMT_MISC is not set
CONFIG_PARPORT=m
CONFIG_PARPORT_PC=m
# CONFIG_PARPORT_OTHER is not set
CONFIG_APM=y
# CONFIG_APM_IGNORE_USER_SUSPEND is not set
```

```
CONFIG_APM_DO_ENABLE=y
CONFIG_APM_CPU_IDLE=y
CONFIG_APM_DISPLAY_BLANK=y
CONFIG_APM_IGNORE_SUSPEND_BOUNCE=y
# CONFIG_APM_RTC_IS_GMT is not set
# CONFIG_APM_ALLOW_INTS is not set
# CONFIG_APM_REAL_MODE_POWER_OFF is not set

#
# Plug and Play support
#
# CONFIG_PNP is not set

#
# Block devices
#
CONFIG_BLK_DEV_FD=y
CONFIG_BLK_DEV_IDE=y

#
# Please see Documentation/ide.txt for help/info on IDE drives
#
# CONFIG_BLK_DEV_HD_IDE is not set
CONFIG_BLK_DEV_IDEDISK=y
CONFIG_BLK_DEV_IDECD=y
# CONFIG_BLK_DEV_IDETAPE is not set
# CONFIG_BLK_DEV_IDEFLOPPY is not set
# CONFIG_BLK_DEV_IDESCSI is not set
# CONFIG_BLK_DEV_CMD640 is not set
# CONFIG_BLK_DEV_RZ1000 is not set
CONFIG_BLK_DEV_IDEPCI=y
CONFIG_BLK_DEV_IDEDMA=y
# CONFIG_BLK_DEV_OFFBOARD is not set
CONFIG_IDEDMA_AUTO=y
# CONFIG_IDE_CHIPSETS is not set

#
# Additional Block Devices
#
# CONFIG_BLK_DEV_LOOP is not set
# CONFIG_BLK_DEV_NBD is not set
# CONFIG_BLK_DEV_MD is not set
CONFIG_BLK_DEV_RAM=m
CONFIG_BLK_DEV_RAM_SIZE=4096
# CONFIG_BLK_DEV_XD is not set
# CONFIG_BLK_DEV_DAC960 is not set
CONFIG_PARIDE_PARPORT=m
# CONFIG_PARIDE is not set
# CONFIG_BLK_CPQ_DA is not set
# CONFIG_BLK_DEV_HD is not set

#
```

```
# Networking options
#
CONFIG_PACKET=m
# CONFIG_NETLINK is not set
# CONFIG_FIREWALL is not set
# CONFIG_FILTER is not set
CONFIG_UNIX=y
CONFIG_INET=y
# CONFIG_IP_MULTICAST is not set
# CONFIG_IP_ADVANCED_ROUTER is not set
# CONFIG_IP_PNP is not set
# CONFIG_IP_ROUTER is not set
# CONFIG_NET_IPIP is not set
# CONFIG_NET_IPGRE is not set
# CONFIG_IP_ALIAS is not set
# CONFIG_SYN_COOKIES is not set

#
# (it is safe to leave these untouched)
#
# CONFIG_INET_RARP is not set
CONFIG_SKB_LARGE=y

#
#
#
# CONFIG_IPX is not set
# CONFIG_ATALK is not set

#
# Telephony Support
#
# CONFIG_PHONE is not set
# CONFIG_PHONE_IXJ is not set

#
# SCSI support
#
CONFIG_SCSI=m

#
# SCSI support type (disk, tape, CD-ROM)
#
CONFIG_BLK_DEV_SD=m
# CONFIG_CHR_DEV_ST is not set
# CONFIG_BLK_DEV_SR is not set
CONFIG_CHR_DEV_SG=m

#
# Some SCSI devices (e.g. CD jukebox) support multiple LUNs
#
CONFIG_SCSI_MULTI_LUN=y
```

```
CONFIG_SCSI_CONSTANTS=y
# CONFIG_SCSI_LOGGING is not set

#
# SCSI low-level drivers
#
# CONFIG_SCSI_7000FASST is not set
# CONFIG_SCSI_ACARD is not set
# CONFIG_SCSI_AHA152X is not set
# CONFIG_SCSI_AHA1542 is not set
# CONFIG_SCSI_AHA1740 is not set
# CONFIG_SCSI_AIC7XXX is not set
# CONFIG_SCSI_IPS is not set
# CONFIG_SCSI_ADVANSYS is not set
# CONFIG_SCSI_IN2000 is not set
# CONFIG_SCSI_AM53C974 is not set
# CONFIG_SCSI_MEGARAID is not set
# CONFIG_SCSI_BUSLOGIC is not set
# CONFIG_SCSI_DTC3280 is not set
# CONFIG_SCSI_EATA is not set
# CONFIG_SCSI_EATA_DMA is not set
# CONFIG_SCSI_EATA_PIO is not set
# CONFIG_SCSI_FUTURE_DOMAIN is not set
# CONFIG_SCSI_GDTH is not set
# CONFIG_SCSI_GENERIC_NCR5380 is not set
# CONFIG_SCSI_INITIO is not set
# CONFIG_SCSI_INIA100 is not set
CONFIG_SCSI_PPA=m
CONFIG_SCSI_IMM=m
CONFIG_SCSI_IZIP_EPP16=y
CONFIG_SCSI_IZIP_SLOW_CTR=y
# CONFIG_SCSI_NCR53C406A is not set
# CONFIG_SCSI_SYM53C416 is not set
# CONFIG_SCSI_SIM710 is not set
# CONFIG_SCSI_NCR53C7xx is not set
# CONFIG_SCSI_NCR53C8XX is not set
# CONFIG_SCSI_SYM53C8XX is not set
# CONFIG_SCSI_PAS16 is not set
# CONFIG_SCSI_PCI2000 is not set
# CONFIG_SCSI_PCI2220I is not set
# CONFIG_SCSI_PSI240I is not set
# CONFIG_SCSI_QLOGIC_FAS is not set
# CONFIG_SCSI_QLOGIC_ISP is not set
# CONFIG_SCSI_QLOGIC_FC is not set
# CONFIG_SCSI_SEAGATE is not set
# CONFIG_SCSI_DC390T is not set
# CONFIG_SCSI_T128 is not set
# CONFIG_SCSI_U14_34F is not set
# CONFIG_SCSI_ULTRASTOR is not set

#
# I2O device support
```

```
#
# CONFIG_I2O is not set
# CONFIG_I2O_PCI is not set
# CONFIG_I2O_BLOCK is not set
# CONFIG_I2O_SCSI is not set

#
# Network device support
#
CONFIG_NETDEVICES=y

#
# ARCnet devices
#
# CONFIG_ARCNET is not set
CONFIG_DUMMY=m
# CONFIG_BONDING is not set
# CONFIG_EQUALIZER is not set
# CONFIG_NET_SB1000 is not set

#
# Ethernet (10 or 100Mbit)
#
# CONFIG_NET_ETHERNET is not set

#
# Ethernet (1000 Mbit)
#
# CONFIG_SK98LIN is not set
# CONFIG_FDDI is not set
# CONFIG_PLIP is not set
CONFIG_PPP=m

#
# CCP compressors for PPP are only built as modules.
#
CONFIG_SLIP=m
CONFIG_SLIP_COMPRESSED=y
CONFIG_SLIP_SMART=y
# CONFIG_SLIP_MODE_SLIP6 is not set
# CONFIG_NET_RADIO is not set

#
# Token ring devices
#
# CONFIG_TR is not set
# CONFIG_NET_FC is not set

#
# Wan interfaces
#
# CONFIG_HOSTESS_SV11 is not set
```

```
# CONFIG_COSA is not set
# CONFIG_SEALEVEL_4021 is not set
# CONFIG_SYNCLINK_SYNCPPP is not set
# CONFIG_LANMEDIA is not set
# CONFIG_COMX is not set
# CONFIG_DLCI is not set
# CONFIG_WAN_DRIVERS is not set
# CONFIG_SBNI is not set

#
# Amateur Radio support
#
# CONFIG_HAMRADIO is not set

#
# IrDA (infrared) support
#
# CONFIG_IRDA is not set

#
# ISDN subsystem
#
# CONFIG_ISDN is not set

#
# Old CD-ROM drivers (not SCSI, not IDE)
#
# CONFIG_CD_NO_IDESCSI is not set

#
# Character devices
#
CONFIG_VT=y
CONFIG_VT_CONSOLE=y
CONFIG_SERIAL=y
# CONFIG_SERIAL_CONSOLE is not set
# CONFIG_SERIAL_EXTENDED is not set
# CONFIG_SERIAL_NONSTANDARD is not set
CONFIG_UNIX98_PTYS=y
CONFIG_UNIX98_PTY_COUNT=256
CONFIG_PRINTER=m
CONFIG_PRINTER_READBACK=y
CONFIG_MOUSE=y

#
# Mice
#
# CONFIG_ATIXL_BUSMOUSE is not set
# CONFIG_BUSMOUSE is not set
# CONFIG_MS_BUSMOUSE is not set
CONFIG_PSMOUSE=y
# CONFIG_82C710_MOUSE is not set
```

```
# CONFIG_PC110_PAD is not set

#
# Joysticks
#
# CONFIG_JOYSTICK is not set
# CONFIG_QIC02_TAPE is not set
# CONFIG_WATCHDOG is not set
# CONFIG_NVRAM is not set
# CONFIG_RTC is not set

#
# Video For Linux
#
# CONFIG_VIDEO_DEV is not set
# CONFIG_DTLK is not set

#
# Ftape, the floppy tape device driver
#
# CONFIG_FTAPE is not set

#
# Filesystems
#
# CONFIG_QUOTA is not set
CONFIG_AUTOFS_FS=y
# CONFIG_AFFS_FS is not set
# CONFIG_HFS_FS is not set
CONFIG_FAT_FS=m
CONFIG_MSDOS_FS=m
# CONFIG_UMSDOS_FS is not set
CONFIG_VFAT_FS=m
CONFIG_ISO9660_FS=m
# CONFIG_JOLIET is not set
CONFIG_MINIX_FS=m
# CONFIG_NTFS_FS is not set
# CONFIG_HPFS_FS is not set
CONFIG_PROC_FS=y
CONFIG_DEVPTS_FS=y
# CONFIG_ROMFS_FS is not set
CONFIG_EXT2_FS=y
# CONFIG_SYSV_FS is not set
# CONFIG_UFS_FS is not set

#
# Network File Systems
#
# CONFIG_CODA_FS is not set
CONFIG_NFS_FS=m
CONFIG_SUNRPC=m
CONFIG_LOCKD=m
```

```
# CONFIG_SMB_FS is not set
# CONFIG_NCP_FS is not set

#
# Partition Types
#
# CONFIG_BSD_DISKLABEL is not set
# CONFIG_MAC_PARTITION is not set
# CONFIG_SMD_DISKLABEL is not set
# CONFIG_SOLARIS_X86_PARTITION is not set
CONFIG_NLS=y

#
# Native Language Support
#
CONFIG_NLS_DEFAULT="cp437"
CONFIG_NLS_CODEPAGE_437=m
# CONFIG_NLS_CODEPAGE_737 is not set
# CONFIG_NLS_CODEPAGE_775 is not set
CONFIG_NLS_CODEPAGE_850=m
# CONFIG_NLS_CODEPAGE_852 is not set
# CONFIG_NLS_CODEPAGE_855 is not set
# CONFIG_NLS_CODEPAGE_857 is not set
# CONFIG_NLS_CODEPAGE_860 is not set
# CONFIG_NLS_CODEPAGE_861 is not set
# CONFIG_NLS_CODEPAGE_862 is not set
# CONFIG_NLS_CODEPAGE_863 is not set
# CONFIG_NLS_CODEPAGE_864 is not set
# CONFIG_NLS_CODEPAGE_865 is not set
# CONFIG_NLS_CODEPAGE_866 is not set
# CONFIG_NLS_CODEPAGE_869 is not set
# CONFIG_NLS_CODEPAGE_874 is not set
# CONFIG_NLS_CODEPAGE_932 is not set
# CONFIG_NLS_CODEPAGE_936 is not set
# CONFIG_NLS_CODEPAGE_949 is not set
# CONFIG_NLS_CODEPAGE_950 is not set
CONFIG_NLS_ISO8859_1=m
# CONFIG_NLS_ISO8859_2 is not set
# CONFIG_NLS_ISO8859_3 is not set
# CONFIG_NLS_ISO8859_4 is not set
# CONFIG_NLS_ISO8859_5 is not set
# CONFIG_NLS_ISO8859_6 is not set
# CONFIG_NLS_ISO8859_7 is not set
# CONFIG_NLS_ISO8859_8 is not set
# CONFIG_NLS_ISO8859_9 is not set
# CONFIG_NLS_ISO8859_14 is not set
CONFIG_NLS_ISO8859_15=m
# CONFIG_NLS_KOI8_R is not set

#
# Console drivers
#
```

```
CONFIG_VGA_CONSOLE=y
# CONFIG_VIDEO_SELECT is not set

#
# Sound
#
CONFIG_SOUND=m
# CONFIG_SOUND_ES1370 is not set
# CONFIG_SOUND_ES1371 is not set
# CONFIG_SOUND_MAESTRO is not set
# CONFIG_SOUND_ICH is not set
# CONFIG_SOUND_SONICVIBES is not set
# CONFIG_SOUND_TRIDENT is not set
# CONFIG_SOUND_MSNDCLAS is not set
# CONFIG_SOUND_MSNDPIN is not set
CONFIG_SOUND_OSS=m
# CONFIG_SOUND_PAS is not set
CONFIG_SOUND_SB=m
CONFIG_SOUND_GUS=m
# CONFIG_GUS16 is not set
# CONFIG_GUSMAX is not set
CONFIG_SOUND_MPU401=m
# CONFIG_SOUND_PSS is not set
CONFIG_SOUND_MSS=m
# CONFIG_SOUND_SSCAPE is not set
# CONFIG_SOUND_TRIX is not set
# CONFIG_SOUND_VIA82CXXX is not set
# CONFIG_SOUND_MAD16 is not set
# CONFIG_SOUND_WAVEFRONT is not set
# CONFIG_SOUND_CS4232 is not set
CONFIG_SOUND_OPL3SA2=m
# CONFIG_SOUND_MAUI is not set
# CONFIG_SOUND_SGALAXY is not set
CONFIG_SOUND_OPL3SA1=m
# CONFIG_SOUND_SOFTOSS is not set
CONFIG_SOUND_YM3812=m
# CONFIG_SOUND_VMIDI is not set
# CONFIG_SOUND_UART6850 is not set
# CONFIG_SOUND_NM256 is not set
# CONFIG_SOUND_YMPCI is not set

#
# Additional low level sound drivers
#
# CONFIG_LOWLEVEL_SOUND is not set

#
# Kernel hacking
#
# CONFIG_MAGIC_SYSRQ is not set
```

76 Mpeg

76.1 mpeg_play

Files:
/usr/local/bin/{mpeg_play, mpeg_save}

I compiled from the source *mpeg_play-2.3-src.tar.gz* obtained somewhere from the Internet. The Makefile needs some minor adjustments.

76.2 libmpeg

Files:
/usr/local/bin/mpeg
*/usr/local/lib/libmpeg.so**
/usr/local/include/mpeg.h
/usr/local/man/cat1/mpeg.1.gz

Sources obtainable at:
`ftp://sunsite.unc.edu/pub/Linux/libs/graphics/libIMPlugIn-1.2-elf.tgz`
`ftp://ftp.mni.mcgill.ca/pub/mpeg/`

The sunsite sources also contains jpeg, tiff, and other libraries. This is a precompiled distribution, I extracted only the files listed above. The sources are also available from the site at McGill (see above). From this file I extracted the header file *mpeg.h*.

76.3 mpegtv

Files:
usr/X11/bin/{mtv, mtvp}
usr/X11/lib/libpthread-mpegtv.so.0.7
usr/X11/man/man1/mtv.1.gz
*usr/doc/mpegtv/**

Sources obtainable at:
`http://www.mpegtv.com`

You need to get *mtv-1.1.0.7.tar.gz* and *libpthread-mpegtv.so.0.gz*. The first archive contains the binaries and a few documentation files. I make everything into a Slackware package, see location of the files above. After installation, don't forget to run `ldconfig` to pick up the library.

76.4 mpeg2vidcodec

Files:
*/usr/local/mpeg2vidcodec/**

Sources obtainable at:
`http://metalab.unc.edu/pub/Linux/distributions/redhat/contrib/libc6/SRPMS/`
`mpeg2vidcodec-1.2-1.src.rpm`

Compile the sources, and install the two binaries, all the files in the *par* directory, and all the documentation. I wrote two short man pages simply telling the user to read the files in */usr/local/mpeg2vidcodec/doc*. I also wrote a short note on how to use these programs to concatenate several MPEG files of identical picture size. For this I also wrote a shell script. Finally link the binaries and man pages from */usr/local*.

The instructions

These are instructions on how to concatenate several MPEG files of identical frame size. THE AUDIO TRACK WILL BE LOST!

Make an empty directory. Place your mpg files into it.

First use the decoder to get the .U .V .Y files. In the example I assume you have three source mpg files you want to concatenate. All the resulting .U .V .Y files have to be consecutively numbered. So start out with the first file, and use the prefix *a*, and decode the second file, use the prefix *b*. I wrote a script that you can run now, to get the *b* files renamed to fit the *a* sequence. Run that script now. Now decode the third file, with *b prefix*, and then add it to the *a* series as well.

```
mpeg2decode -b 1.mpg -o0 a%d
mpeg2decode -b 2.mpg -o0 b%d
mpeg2decode_add_b_to_a_series
mpeg2decode -b 3.mpg -o0 b%d
mpeg2decode_add_b_to_a_series
```

You will now also know the total number of frames. At this stage use xv to grab a window of `mpeg_play 1.mpg` to get the size of the movie. Copy *par/MPEG-1.par* to the current directory, and edit the following lines.

```
a%d       /* name of source files */
879       /* number of frames */
160       /* horizontal_size */
128       /* vertical_size */
160       /* display_horizontal_size */
128       /* display_vertical_size */
```

Finally encode the whole thing. Ignore the "vbv_decode overflow" errors. Your resulting file is called *a.mpg* in the example.

```
mpeg2encode MPEG-1.par a.mpg
```

The script mpeg2decode_add_b_to_a_series

```
oldnumber=‘ls a*.U | sed -e ’s/a//’ -e ’s/..$//’| sort -k 1n | tail -n 1‘
oldnumber=$[oldnumber+1]
echo oldnumber=$oldnumber
for i in b*; do
j_old=‘echo $i | sed -e ’s/b//’ -e ’s/..$//’‘
j_new=$[j_old+oldnumber]
mv $i a${j_new}‘echo $i|sed -e ’s/.*\(..\)$/\1/’‘
echo -n "."
```

```
done
echo " done"
oldnumber=‘ls a*.U | sed -e ’s/a//’ -e ’s/..$//’| sort -k 1n | tail -n 1‘
oldnumber=$[oldnumber+1]
echo newnumber=$oldnumber
```

77 MuPAD

Files:
*/usr/local/MuPAD/**
/etc/profile

Sources obtainable at:
`http://www.sciface.com/`
`ftp://www.sciface.com/`
`http://math-www.uni-paderborn.de/MuPAD/`

MuPAD is a computer algebra system developed at the Universität Paderborn in Germany. You'll need two archives, a set of binaries and the common shared files: *bin_i386_142.tgz* and *share_142.tgz*. Make a directory */usr/local/MuPAD* and unpack the sources from within. It uses about 17 megabytes of disk space. Now you have two options.

1. Modify the system wide initialization file. You'll need to add the following.

   ```
   #
   MuPAD_ROOT_PATH=/usr/local/MuPAD
   FONT_PATH=$MuPAD_ROOT_PATH/share/doc/hytex/fonts
   PATH=$PATH:$MuPAD_ROOT_PATH/share/bin
   MANPATH=$MANPATH:$MuPAD_ROOT_PATH/share/doc/man
   PAGER=’less -s -w’
   export MuPAD_ROOT_PATH FONT_PATH PATH PAGER
   #
   ```

2. Or the second option is: Write a wrapper script that calls the actual binaries, and set the paths inside those wrapper scripts.

Either way, I suggest you gzip the man pages in */usr/local/MuPAD/share/doc/man/man?*. Notice that I extended the MANPATH as explained above.

MuPAD needs Xview

MuPAD needs the latest version of `xview` installed, find this on any Linux ftp site. Notice that cut-and-paste works differently for xview applications. I have it set so that F10 is cut, F11 is copy, and F12 is paste. Selection is done by highlighting with the mouse. For this to work you need to add a few lines to your *.Xmodmap* file.

```
! *** Installed by xview3L5 ***
! F8=Help  (move pointer on panel, press F1 to show help on the item)
! F9=Find  (after having selected some text, press F2 to do a search)
! F10=Cut   (select text, press F3 to move text into clipboard)
! F11=Copy  (select text, press F4 to copy text into clipboard)
! F12=Paste (insert text from clipboard at caret position)
keysym F8 = Help
keysym F9 = F19
keysym F10 = F20
keysym F11 = F16
keysym F12 = F18
```

Extra documentation

I downloaded many of the pages under *http://www.sciface.com/support/*, in particular the directory *papers*. These pages need to be adjusted so that the paths are no longer absolute, use sed of something. Then I downloaded also the official documentation kit, it's *html_help.tgz*, this contains a *README* file, and the actual archive *mupad_html_help.tgz*. I like to keep my HTML files gzipped, so I installed as follows.

```
cd /usr/local/MuPAD
tar -zxvf /usr/src/MuPAD/mupad_html_help.tgz
cd mupad_html_help
for i in *.html; do echo "s/$i/$i.gz/g"; done > sed.f
for i in *.html; do sed -f sed.f $i > foo && cat foo > $i; echo $i; done
rm -f foo sed.f
```

Tis actually changes one link incorrectly, the one to *http://www.sciface.com/index.html* in the file *index.html*. Fix that, if you wish. Link this documentation like any other HTML formatted documentation from within */usr/local/html/index.html*. The documentation takes up about 4 megabytes.

Registration

MuPAD as downloaded is a demo version. Registration is free. The easiest way to register is to fill in the form on the web site. I did that and got an immediate confirmation with the license a day later. This is a code that one enters once while running MuPAD, and it removes the memory restriction permanently. Run `mupad` as root and issue the command `register("your code name", "your license code");`. That does the trick.

78 Named

Files:
/var/named/{*root.cache, 0.0.127, localhost*}
/etc/{*named.conf, resolve.conf, networks, HOSTNAME, hosts, host.conf*}
/etc/rc.d/{*rc.inet1, rc.inet2*}

This is the configuration for the domain name server. I have it set up so that the `named` daemon caches addresses it has already looked up. Unfortunately it will lose its memory when

the system goes down. It also allows for reverse lookup of local domain names. Read the man page `named(8)` for info about the cryptic entries of the configuration files. Up to version 4.9 of *bind* the configuration file was */etc/named.boot*, which has by now (version 8.8.2) been switched to */etc/named.conf*. Of course, the syntax changed also, and there is a utility to change an old-syntax file to a new one, it is */sbin/named-bootconf*, or with */sbin/named-bootconf.pl* for the *Perl* version. the file list above is far from exhaustive, I only list those files that needed modification.

/etc/named.conf

```
// generated by named-bootconf.pl from the old named.boot
//
// First a list of forwarders for recursive queries. This should form a cache
// on the queried machines.
//
//
// nameserver.UTCC.UTK.Edu 128.169.202.79
// t-online.de 194.25.2.129
//
options {
directory "/var/named";
forwarders {
194.25.2.129;
128.169.202.79;
};
/*
 * If there is a firewall between you and nameservers you want
 * to talk to, you might need to uncomment the query-source
 * directive below.  Previous versions of BIND always asked
 * questions using port 53, but BIND 8.1 uses an unprivileged
 * port by default.
 */
// query-source address * port 53;
};

//
// boot file for name server
//
// This only specifies a number of root servers listed in /var/named/named.ca.
// Furthermore, named will store addresses it has already looked up in memory.
// It unfortunately means named doesn't remember anything if the system is
// rebooted.
//
// type     domain master file
//
zone "." {
type hint;
file "root.cache";
};

//
// This allows for reverse lookup of a domain name given the numeric address.
```

```
// It only gives information about the local host.
//
zone "0.0.127.IN-ADDR.ARPA" {
type master;
file "0.0.127";
};

//
// This allows for lookup of the local host.
// It only gives information about the local host.
//
zone "localhost" {
        type master;
        file "localhost";
};
```

root.cache

```
;
; /var/named/root.cache cache file for localhost
; Origin is .
;
;       This file holds the information on root name servers needed to
;       initialize cache of Internet domain name servers
;       (e.g. reference this file in the "cache  .  <file>"
;       configuration file of BIND domain name servers).
;
;       This file is made available by InterNIC registration services
;       under anonymous FTP as
;           file                /domain/named.root
;           on server           FTP.RS.INTERNIC.NET
;       -OR- under Gopher at    RS.INTERNIC.NET
;           under menu          InterNIC Registration Services (NSI)
;              submenu          InterNIC Registration Archives
;           file                named.root
;
;       last update:    Aug 22, 1997
;       related version of root zone:   1997082200
;
;
; formerly NS.INTERNIC.NET
;
.                        3600000  IN  NS    A.ROOT-SERVERS.NET.
A.ROOT-SERVERS.NET.      3600000      A     198.41.0.4
;
; formerly NS1.ISI.EDU
;
.                        3600000      NS    B.ROOT-SERVERS.NET.
B.ROOT-SERVERS.NET.      3600000      A     128.9.0.107
;
; formerly C.PSI.NET
```

```
;
.                       3600000      NS    C.ROOT-SERVERS.NET.
C.ROOT-SERVERS.NET.     3600000      A     192.33.4.12
;
; formerly TERP.UMD.EDU
;
.                       3600000      NS    D.ROOT-SERVERS.NET.
D.ROOT-SERVERS.NET.     3600000      A     128.8.10.90
;
; formerly NS.NASA.GOV
;
.                       3600000      NS    E.ROOT-SERVERS.NET.
E.ROOT-SERVERS.NET.     3600000      A     192.203.230.10
;
; formerly NS.ISC.ORG
;
.                       3600000      NS    F.ROOT-SERVERS.NET.
F.ROOT-SERVERS.NET.     3600000      A     192.5.5.241
;
; formerly NS.NIC.DDN.MIL
;
.                       3600000      NS    G.ROOT-SERVERS.NET.
G.ROOT-SERVERS.NET.     3600000      A     192.112.36.4
;
; formerly AOS.ARL.ARMY.MIL
;
.                       3600000      NS    H.ROOT-SERVERS.NET.
H.ROOT-SERVERS.NET.     3600000      A     128.63.2.53
;
; formerly NIC.NORDU.NET
;
.                       3600000      NS    I.ROOT-SERVERS.NET.
I.ROOT-SERVERS.NET.     3600000      A     192.36.148.17
;
; temporarily housed at NSI (InterNIC)
;
.                       3600000      NS    J.ROOT-SERVERS.NET.
J.ROOT-SERVERS.NET.     3600000      A     198.41.0.10
;
; housed in LINX, operated by RIPE NCC
;
.                       3600000      NS    K.ROOT-SERVERS.NET.
K.ROOT-SERVERS.NET.     3600000      A     193.0.14.129
;
; temporarily housed at ISI (IANA)
;
.                       3600000      NS    L.ROOT-SERVERS.NET.
L.ROOT-SERVERS.NET.     3600000      A     198.32.64.12
;
; housed in Japan, operated by WIDE
;
.                       3600000      NS    M.ROOT-SERVERS.NET.
```

```
M.ROOT-SERVERS.NET.      3600000      A     202.12.27.33
; End of File
```

0.0.127

```
;
; /var/named/named.0.0.127 reverse mapping of 127.0.0
; Origin is 0.0.127.IN-ADDR.ARPA.
;
$TTL 3600000
;
@ IN SOA localhost root.localhost 1 360000 3600 3600000 604800
;
; localhost       ; primary name server
; root.localhost  ; e-mail address of contact
; 1         ; version number
; 360000          ; refresh:  100 hours
; 3600          ; retry:      1 hour
; 3600000          ; expire:    42 days
; 604800          ; minimum:    1 week
;
IN NS localhost.
1 IN PTR localhost.
```

/etc/networks

```
#
# networks This file describes a number of netname-to-address
# mappings for the TCP/IP subsystem.  It is mostly
# used at boot time, when no name servers are running.
#

loopback 127.0.0.0
localnet 127.0.0.0

# End of networks.
```

/etc/HOSTNAME

```
tosca
```

/etc/hosts

```
#
# hosts This file describes a number of hostname-to-address
# mappings for the TCP/IP subsystem.  It is mostly
# used at boot time, when no name servers are running.
# On small systems, this file can be used instead of a
# "named" name server.  Just add the names, addresses
# and any aliases to this file...
#
```

```
# By the way, Arnt Gulbrandsen <agulbra@nvg.unit.no> says that 127.0.0.1
# should NEVER be named with the name of the machine.  It causes problems
# for some (stupid) programs, irc and reputedly talk. :^)
#

# For loopbacking.
127.0.0.1               localhost
127.0.0.1               tosca.localnet          tosca

# End of original hosts as distributed by slackware.
```

/etc/host.conf

```
order hosts,bind
multi on
```

/etc/rc.d/rc.inet1

```
#! /bin/sh
#
# rc.inet1 This shell script boots up the base INET system.
#
# Version: @(#)/etc/rc.d/rc.inet1 2.00 10/06/1999
#

HOSTNAME=‘cat /etc/HOSTNAME‘

# Attach the loopback device.
/sbin/ifconfig lo 127.0.0.1
/sbin/route add -net 127.0.0.0 netmask 255.0.0.0 lo

# IF YOU HAVE AN ETHERNET CONNECTION, use these lines below to configure the
# eth0 interface.

# Edit these values to set up a static IP address:
IPADDR="127.0.0.1" # REPLACE with YOUR IP address!
NETMASK="255.255.255.0" # REPLACE with YOUR netmask!
NETWORK="127.0.0.0" # REPLACE with YOUR network address!
BROADCAST="" # REPLACE with YOUR broadcast address, if you
# have one. If not, leave blank and edit below.
GATEWAY="" # REPLACE with YOUR gateway address!

# To use DHCP instead of a static IP, set this value to "yes":
DHCP="no"            # Use DHCP ("yes" or "no")

# OK, time to set up the interface:
if [ "$DHCP" = "yes" ]; then # use DHCP to set everything up:
  echo "Attempting to configure eth0 by contacting a DHCP server..."
  /sbin/dhcpcd
elif [ ! "$IPADDR" = "127.0.0.1" ]; then # set up IP statically:
  # Set up the ethernet card:
  echo "Configuring eth0 as ${IPADDR}..."
```

```
  /sbin/ifconfig eth0 ${IPADDR} broadcast ${BROADCAST} netmask ${NETMASK}

  # If that didn't succeed, give the system administrator some hints:
  if [ ! $? = 0 ]; then
    cat << EOF
Your ethernet card was not initialized properly.  Here are some reasons why this
may have happened, and the solutions:
1. Your kernel does not contain support for your card.  Including all the
   network drivers in a Linux kernel can make it too large to even boot, and
   sometimes including extra drivers can cause system hangs.  To support your
   ethernet, either edit /etc/rc.d/rc.modules to load the support at boottime,
   or compile and install a kernel that contains support.
2. You don't have an ethernet card, in which case you should comment out this
   section of /etc/rc.d/rc.inet1.  (Unless you don't mind seeing this error...)
EOF
  fi

  # Older kernel versions need this to set up the eth0 routing table:
  KVERSION=`uname -r | cut -f 1,2 -d .`
  if [ "$KVERSION" = "1.0" -o "$KVERSION" = "1.1" \
   -o "$KVERSION" = "1.2" -o "$KVERSION" = "2.0" -o "$KVERSION" = "" ]; then
    /sbin/route add -net ${NETWORK} netmask ${NETMASK} eth0
  fi

  # If there is a gateway defined, then set it up:
  if [ ! "$GATEWAY" = "" ]; then
    /sbin/route add default gw ${GATEWAY} netmask 0.0.0.0 metric 1
  fi
fi

# End of rc.inet1
```

/etc/rc.d/rc.inet2

Only the parts that relate to `named`. This file also starts other daemons.

```
#!/bin/sh
#
# rc.inet2 This shell script boots up the entire INET system.
# Note, that when this script is used to also fire
# up any important remote NFS disks (like the /usr
# distribution), care must be taken to actually
# have all the needed binaries online _now_ ...
#
# Author: Fred N. van Kempen, <waltje@uwalt.nl.mugnet.org>
# Modified for Slackware by Patrick Volkerding <volkerdi@slackware.com>
#

# Some constants:
NET="/usr/sbin"
IN_SERV="lpd"
LPSPOOL="/var/spool/lpd"
```

```
# If we see IPv4 packet forwarding support in the kernel, we will turn it on.
# This was the default for 2.0.x kernels, but with recent kernels it must be
# activated through a file in /proc.  IPv4 packet forwarding support is
# required if you plan to use your Linux machine as a router or firewall.
# If you don't want your Linux machine to forward packets, change the 1 below
# to a 0.
IPV4_FORWARD=1
if [ -f /proc/sys/net/ipv4/ip_forward ]; then
  if [ "$IPV4_FORWARD" = "1" ]; then
    echo "Activating IPv4 packet forwarding..."
    echo 1 > /proc/sys/net/ipv4/ip_forward
  else
    echo "Disabling IPv4 packet forwarding..."
    echo 0 > /proc/sys/net/ipv4/ip_forward
  fi
fi

# When using IPv4 packet forwarding, you will also get the rp_filter, which
# automatically rejects incoming packets if the routing table entry for their
# source address doesn't match the network interface they're arriving on.  This
# has security advantages because it prevents the so-called IP spoofing,
# however it can pose problems if you use asymmetric routing (packets from you
# to a host take a different path than packets from that host to you) or if
# you operate a non-routing host which has several IP addresses on different
# interfaces.  To turn rp_filter off, uncomment the lines below:
# if [ -r /proc/sys/net/ipv4/conf/all/rp_filter ]; then
#   echo "Disabling rp_filter..."
#   echo 0 > /proc/sys/net/ipv4/conf/all/rp_filter
# fi

# Start the SUN RPC Portmapper:
if [ -x /sbin/rpc.portmap ]; then
  echo "Starting /sbin/rpc.portmap..."
  /sbin/rpc.portmap
fi

# At this point, we are ready to talk to The World...

# Mount NFS filesystems:
echo "Mounting remote file systems..."
/sbin/mount -a -t nfs        # This may be our /usr runtime!!!
# Show the mounted volumes:
/sbin/mount -v -t nfs

# Begin a list of started daemons:
echo -n "Starting daemons: "

# Start the SYSLOGD/KLOGD daemons:
if [ -x ${NET}/syslogd ]; then
  echo -n " syslogd"
  ${NET}/syslogd
```

```
  sleep 1 # prevent syslogd/klogd race condition on SMP kernels
  echo -n " klogd"
  # '-c 3' = display level 'error' or higher messages on console
  ${NET}/klogd -c 3
fi

# Start the INET SuperServer:
if [ -x ${NET}/inetd ]; then
  echo -n " inetd"
  ${NET}/inetd
else
  echo
  echo "WARNING:  ${NET}/inetd not found."
  echo -n "Continuing daemon loading: "
fi

# Look for sshd in the two most common locations (compiled with --prefix=/usr
# or with --prefix=/usr/local) and if we find it, start it up
#if [ -x /usr/local/sbin/sshd ]; then
#  echo -n " sshd"
#  /usr/local/sbin/sshd
#elif [ -x /usr/sbin/sshd ]; then
#  echo -n " sshd"
#  /usr/sbin/sshd
#fi

# Option number one:  Start the NAMED/BIND name server as root.
# Note that BIND has a somewhat spotty history in terms of security,
# but they recommend running it as root, and if you don't, not all
# the features (especially when attached to a dynamic IP such as
# through PPP) will neccessary work correctly).
if [ -x ${NET}/named ]; then
  echo -n " named"
  ${NET}/named
fi

# # Option number two:  Start the NAMED/BIND name server as user daemon.
# # This is another way to start BIND for the more paranoid.  By starting it as
# # user daemon, group daemon, if there does turn out to be a vulnerability,
# # the attacker can at best gain user deamon access to your machine, which is
# # a considerably better situation than if they gain root access.  Note that
# # for this option to work at all, you'll need to change the ownership of the
# # directory /var/named to daemon like this: chown -R daemon.daemon /var/named
# # If you don't have this directory, you'll have to make it first.
# # Also, for the *really* paranoid, you can read this document for
# # instructions on running BIND in a "chroot jail":
# #   /usr/doc/Linux-HOWTOs/Chroot-BIND-HOWTO
# if [ -x ${NET}/named ]; then
#   echo -n " named"
#   ${NET}/named -u daemon -g daemon
# fi
```

```
# # Start the ROUTEd server:
# if [ -x ${NET}/routed ]; then
#   echo -n " routed"
#   ${NET}/routed -g -s
# fi

# # Start the RWHO server:
# if [ -x ${NET}/rwhod ]; then
#   echo -n " rwhod"
#   ${NET}/rwhod
# fi

# Start the various INET servers:
for server in ${IN_SERV} ; do
  if [ -x ${NET}/${server} ]; then
    echo -n " ${server}"
    ${NET}/${server}
  fi
done

# # Setting up NIS:
# # (NOTE: For detailed information about setting up NIS, see the documentation
# # in /usr/doc/yp-tools, /usr/doc/ypbind, and /usr/doc/ypserv)
# #
# # First, we must set the NIS domainname.  NOTE: this is not
# # necessarily the same as your DNS domainname, set in
# # /etc/resolv.conf!  The NIS domainname is the name of a domain
# # served by your NIS server.
#
# if [ -r /etc/defaultdomain ]; then
#   nisdomainname `cat /etc/defaultdomain`
# fi
#
# # Then, we start up ypbind.  It will use broadcast to find a server.
#
# if [ -d /var/yp ] ; then
#   echo -n " ypbind"
#   ${NET}/ypbind
# fi
#
# # If you are the NIS master server for the NIS domain, then
# # you must run rpc.yppasswdd, which is the RPC server that
# # lets users change their passwords.
#
# if [ -x ${NET}/rpc.yppasswdd ]; then
#   echo -n " yppasswdd"
#   ${NET}/rpc.yppasswdd
# fi

# # Start the various SUN RPC servers:
if [ -x /sbin/rpc.portmap ]; then
  # Start the NFS server daemons.
```

```
  if [ -x ${NET}/rpc.mountd ]; then
    echo -n " mountd"
    ${NET}/rpc.mountd
  fi
  if [ -x ${NET}/rpc.nfsd ]; then
    echo -n " nfsd"
    ${NET}/rpc.nfsd
  fi
#  # Fire up the PC-NFS daemon(s):
#  if [ -x ${NET}/rpc.pcnfsd ]; then
#    echo -n " pcnfsd"
#    ${NET}/rpc.pcnfsd ${LPSPOOL}
#  fi
#  if [ -x ${NET}/rpc.bwnfsd ]; then
#    echo -n " bwnfsd"
#    ${NET}/rpc.bwnfsd ${LPSPOOL}
#  fi
fi # Done starting various SUN RPC servers.

# The 'echo' below will put a carriage return at the end
# of the list of started servers.
echo

# Done!
```

/etc/resolv.conf

```
nameserver 127.0.0.1
search localnet
```

/etc/nsswitch.conf

```
#
# /etc/nsswitch.conf
#
# An example Name Service Switch config file. This file should be
# sorted with the most-used services at the beginning.
#
# The entry '[NOTFOUND=return]' means that the search for an
# entry should stop if the search in the previous entry turned
# up nothing. Note that if the search failed due to some other reason
# (like no NIS server responding) then the search continues with the
# next entry.
#
# Legal entries are:
#
#      nisplus or nis+         Use NIS+ (NIS version 3)
#      nis or yp               Use NIS (NIS version 2), also called YP
#      dns                     Use DNS (Domain Name Service)
#      files                   Use the local files
#      [NOTFOUND=return]       Stop searching if not found so far
```

```
#

# passwd:  db files nis
# shadow:  db files nis
# group:   db files nis

passwd:  compat
group:   compat

hosts:   files dns
networks:       files dns

services:       db files
protocols:      db files
rpc:     db files
ethers:  db files
netmasks:       files
netgroup:       files
bootparams:     files

automount:      files
aliases:        files
```

79 NcFtp

Files:
/usr/local/bin/ncftp
/usr/local/man/man1/ncftp.1.gz
/usr/local/html/NCFTP_GUIDE.html.gz

Sources obtainable at:
`http://www.ncftp.com/`
`http://frodo.colorado.edu/research/checkpoint/NCFTP_Guide.html` (for the users' guide)

Start with `CFLAGS="-O2"./configure`. This will hang because it tries to check whether we can download something from the Internet. So you either wait for the time-out, or you get on-line for the configure step. Then run `make` and after it's done, `make install`. Don't forget to compress the man page.

If you also get the guide mentioned in the Sources section, then compress it and install it into */usr/local/html*, and make an entry into the file */usr/local/html/index.html*.

80 Nedit

Files:
/usr/local/nedit/bin/nedit
/usr/local/nedit/man/man1/nedit.1.gz

/usr/local/nedit/docs/{nedit.doc, README, ReleaseNotes}
~/{.nedit, .neditdb}

Sources obtainable at:
`http://www.nedit.org`

Get the binary distribution, as of this writing it is *nedit-5.1.1-linux-glibc.tar.gz*. Unpack the tar-ball, it creates its own directory, strip the binary *nedit*. I edited the man page to give the user the directory where the documentation is stored. I did not install *nc*, the network client version.

81 NetPBM

Files:
*/usr/local/netpbm/**

Sources obtainable at:
`http://uarchive.wustl.edu/graphics/graphics/packages/NetPBM`

The sources compile with only minor configuration, however they produce much duplicated code, as they do not produce shared libraries. I have therefore modified the installation procedure to provide shared libraries, that brought the disk use of the binaries down from 3.9 Megabytes to less than 1.5 Megabyte (including the shared libraries). Here are first the instructions without shared library support. Change the following files.
Imakefile

Add the line below near the top of the file.

```
CDEBUGFLAGS = -O3 -fomit-frame-pointer
```

Also, many lines begin with a few spaces instead of a tab, this will cause an error when running `make` later, so use `sed` or something like it to fix this problem now.

Pbmplus.tmpl

I want to use the installed TIFF library (which in turn needs the JPEG, the Z compression, and the math library), and I want to install into */usr/local/netpbm*. So add the following lines near the beginning of the file.

```
#define LibTiff /usr/local/lib/libtiff.so /usr/local/lib/libjpeg.so /usr/local/l
ib/libz.so -lm
#define PbmplusDir    /usr/local/netpbm/
#define PbmplusBinDir bin
#define PbmplusManDir man
#define PbmplusLibDir lib
```

Installation

```
tar -zxf netpbm-1mar1994.tar.gz
cd netpbm
xmkmf
make SUBDIRS='./pbm ./pgm ./ppm ./pnm' Makefiles
make SUBDIRS='./pbm ./pgm ./ppm ./pnm'
make SUBDIRS='./pbm ./pgm ./ppm ./pnm' install
make SUBDIRS='./pbm ./pgm ./ppm ./pnm' install.man
```

Now compress the man pages, and then link the whole thing from */usr/local/*{*bin, man/man1, man/man3, man/man5*}. I chose not to link *pbmtog3* and *g3topbm*, because the **mgetty+sendfax** package already has improved versions of these two converters.

Shared libraries

The instructions are just about the same, but you need to change a few more files and you need to manually install the shared libraries.

```
tar -zxf netpbm-1mar1994.tar.gz
zcat netpbm-elf-patch.gz | patch -p0
cd netpbm
xmkmf
make -f dynamic.makefile
make SUBDIRS='./pbm ./pgm ./ppm ./pnm' Makefiles
make SUBDIRS='./pbm ./pgm ./ppm ./pnm'
make SUBDIRS='./pbm ./pgm ./ppm ./pnm' install
make SUBDIRS='./pbm ./pgm ./ppm ./pnm' install.man
mkdir /usr/local/netpbm/lib
cp `find . -name \*.so.1` /usr/local/netpbm/lib/
```

Now link everything from */usr/local*, and then run `ldconfig -v` so that the new dynamic libraries are actually picked up. For this you need to add the line `/usr/local/netpbm/lib` to the file */etc/ld.so.conf*.

Pbmplus.tmpl

If you want shared libraries, then also add the following lines.

```
#define LibPbm $(PBMDIR)/libpbm.so.1
#define LibPgm $(PGMDIR)/libpgm.so.1
#define LibPpm $(PPMDIR)/libppm.so.1
#define LibPnm $(PNMDIR)/libpnm.so.1
```

pbm/Imakefile

Change the following line.

```
#define LibPbm libpbm.so.1
```

pgm/Imakefile

Change the following line.

```
#define LibPbm libpgm.so.1
```

ppm/Imakefile

Change the following line.

```
#define LibPbm libppm.so.1
```

pnm/Imakefile

Change the following line.

```
#define LibPbm libpnm.so.1
```

dynamic.makefile

```
PBM_CFLAGS = -fPIC -O3 -fomit-frame-pointer   -I..  -I/usr/X11R6/include    -DFU
NCPROTO=15 -DNARROWPROTO  -DRGB_DB=\"/usr/X11R6/lib/X11/rgb\" -DLIBTIFF

PGM_CFLAGS = -fPIC -O3 -fomit-frame-pointer   -I.. -I../pbm  -I/usr/X11R6/includ
e    -DFUNCPROTO=15 -DNARROWPROTO

PPM_CFLAGS = -fPIC -O3 -fomit-frame-pointer   -I.. -I../pbm -I../pgm  -I/usr/X11
R6/include    -DFUNCPROTO=15 -DNARROWPROTO  -DRGB_DB=\"/usr/X11R6/lib/X11/rgb\"

PNM_CFLAGS = -fPIC -O3 -fomit-frame-pointer   -I.. -I../pbm -I../pgm -I../ppm -I
../libtiff  -I/usr/X11R6/include    -DFUNCPROTO=15 -DNARROWPROTO  -DLIBTIFF

all: pbm/libpbm.so.1 pgm/libpgm.so.1 ppm/libppm.so.1 pnm/libpnm.so.1

pbm/libpbm.so.1:
cd pbm ;\
gcc $(PBM_CFLAGS) -c libpbm1.c -o libpbm1.o ;\
echo "made libpbm1.o" ;\
gcc $(PBM_CFLAGS) -c libpbm2.c -o libpbm2.o ;\
echo "made libpbm2.o" ;\
gcc $(PBM_CFLAGS) -c libpbm3.c -o libpbm3.o ;\
echo "made libpbm3.o" ;\
gcc $(PBM_CFLAGS) -c libpbm4.c -o libpbm4.o ;\
echo "made libpbm4.o" ;\
gcc $(PBM_CFLAGS) -c libpbm5.c -o libpbm5.o ;\
echo "made libpbm5.o" ;\
gcc -shared -Wl,-soname,libpbm.so.1 -o libpbm.so.1 \
libpbm1.o libpbm2.o libpbm3.o libpbm4.o libpbm5.o ;\
echo "made libpbm.so.1"

pgm/libpgm.so.1:
cd pgm ;\
```

```
gcc $(PGM_CFLAGS) -c libpgm1.c -o libpgm1.o ;\
echo "made libpgm1.o" ;\
gcc $(PGM_CFLAGS) -c libpgm2.c -o libpgm2.o ;\
echo "made libpgm2.o" ;\
gcc -shared -Wl,-soname,libpgm.so.1 -o libpgm.so.1 \
libpgm1.o libpgm2.o ;\
echo "made libpgm.so.1"

ppm/libppm.so.1:
cd ppm ;\
gcc $(PPM_CFLAGS) -c libppm1.c -o libppm1.o ;\
echo "made libppm1.o" ;\
gcc $(PPM_CFLAGS) -c libppm2.c -o libppm2.o ;\
echo "made libppm2.o" ;\
gcc $(PPM_CFLAGS) -c libppm3.c -o libppm3.o ;\
echo "made libppm3.o" ;\
gcc $(PPM_CFLAGS) -c libppm4.c -o libppm4.o ;\
echo "made libppm4.o" ;\
gcc $(PPM_CFLAGS) -c libppm5.c -o libppm5.o ;\
echo "made libppm5.o" ;\
gcc $(PPM_CFLAGS) -c bitio.c -o bitio.o ;\
echo "made bitio.o" ;\
gcc -shared -Wl,-soname,libppm.so.1 -o libppm.so.1 \
libppm1.o libppm2.o libppm3.o libppm4.o libppm5.o bitio.o;\
echo "made libppm.so.1"

pnm/libpnm.so.1:
cd pnm ;\
gcc $(PNM_CFLAGS) -c libpnm1.c -o libpnm1.o ;\
echo "made libpnm1.o" ;\
gcc $(PNM_CFLAGS) -c libpnm2.c -o libpnm2.o ;\
echo "made libpnm2.o" ;\
gcc $(PNM_CFLAGS) -c libpnm3.c -o libpnm3.o ;\
echo "made libpnm3.o" ;\
gcc $(PNM_CFLAGS) -c libpnm4.c -o libpnm4.o ;\
echo "made libpnm4.o" ;\
gcc -shared -Wl,-soname,libpnm.so.1 -o libpnm.so.1 \
libpnm1.o libpnm2.o libpnm3.o libpnm4.o ;\
echo "made libpnm.so.1"
```

Version 8.3

Compiles under Linux just with the `make` command, no editing of source files necessary. Also installs by default into */usr/local/netpbm*. This package is contained in the *glib* graphics package of Slackware, so I did not install it separately.

82 Netscape

Files:
Netscape-6 files:

/usr/local/bin/netscape
*/usr/local/netscape/**
~/.mozilla/

Communicator-4.x files:

/usr/bin/netscape
*/usr/lib/netscape/**
/var/X11R6/lib/app-defaults/Netscape
~/nsmail/ ~/.mailcap*
~/.Xdefaults
*~/.netscape/**

Netscape is freely available from `http://www.netscape.com`. Currently there are several version available. I installed the US version release 4.73 as part of Slackware, and Netscape 6.

Netscape Navigator 4.x

This is for non-Slackware installation. Unpack the sources, they create their own directory. In there you find the install script called *ns-install*, run it. I install into */usr/local/netscape/* and then installed a wrapper script which is installed as */usr/local/bin/netscape-4*:

```
#!/bin/sh
MOZILLA_HOME="/usr/local/netscape"
export MOZILLA_HOME
exec ${MOZILLA_HOME}/netscape -name netscape $*
```

You also need to set the preferences inside of Netscape. That is under the options menu you need to specify your return email address and name, your SMTP mail server (`math.vanderbilt.edu`), and your news server (`news.vanderbilt.edu`). Netscape reads the *.mailcap* file in your home directory to specify certain helper applications. Note that the two versions do not change their cache files.

mailcap

```
# This is a simple example mailcap file.
# Lines starting with '#' are comments.

# This maps realaudio files to the realaudio player
audio/x-pn-realaudio; raplayer %s

# This maps all types of audio data (audio/basic, audio/x-aiff,
# etc.) to the viewer 'play_save'.  Note that '%s' means 'put the
# datafile name here when the viewer is executed'.
audio/*; play_save %s

# for g3 faxes
```

```
image/g3fax; viewfax %s

# This maps all types of images (image/gif, image/jpeg, etc.)
# to the viewer 'xv'.
image/*; xv %s

# This maps MPEG video data to the viewer 'mpeg_save'.
# I rigged this so it saves the mpeg first
video/mpeg; mpeg_save %s

# This maps any other video data to the viewer 'xanim_save'.
# I rigged this so it saves the file first
video/*;xanim_save %s; xanim %s >/dev/null 2>&1

#application/postscript; gv_save %s
#application/dvi; xdvi_save %s
application/x-dvi; xdvi %s
application/postscript; ghostview %s
application/pdf; acroread %s
application/x-fig; xfig %s
application/x-tex;xterm -n LESS -T 'LESS (q to quit or ! cat \% > foo.txt to
    save' -e less %s
application/x-latex;xterm -n LESS -T 'LESS (q to quit or ! cat \% > foo.txt to
    save' -e less %s
#mailcap entry added by Netscape Helper
audio/x-wav;wavplay_save %s &>/dev/null
#mailcap entry added by Netscape Helper
midi;midplay %s
#mailcap entry added by Netscape Helper
audio/x-midi;midplay %s
#mailcap entry added by Netscape Helper
audio/x-aiff;/home/mayer/bin/playmidi %s
```

Netscape 6

This now comes with a download program *netscape-i686-pc-linux-gnu-installer.tar.gz*. Unpack it, and cd to the newly created directory, and then run *netscape-installer*. This brings up a dialog box. I used a custom installation, and chose not to install the Messenger, and also not the British spell checker. Then the program proceeds to download the desired components. These get stored temporarily in */tmp/.tmp.xi.0*. I saved them, and put them for later re-installation into */home/ftp/netscape*. In the file *config.inf*, which is part of the download-tar-ball, the URLs of the sources are specified. By changing the URL0 to *ftp://127.0.0.1/netscape/*, and replacing URL1 with the old URL0, the download program will first look for the local versions, and download only when needed. This allows for a re-install without a re-download.

It is not needed to register for the Netcenter during registration, just press the cancel button.

I use a wrapper script because Netscape version 6 dumps a lot of screen output giving information about its progress. Furthermore there is a trace file created in the home directory, which I don't need.

```
#!/bin/sh
```

```
xterm -iconic -T "Netscape output" -fn fixed \
      -exec /usr/local/netscape/netscape $*
/bin/rm ${HOME}/plugin130_01.trace
```

82.1 Fonts in Netscape

This is somewhat of a mess. First of all, Netscape substitutes fonts it does not have, and it scales fonts to sizes it does not have. I decided to disable the fonts scaling in the preferences menu, because I rather have prettier fonts of somewhat wrong size, than uglier fonts of correct size.

Now, the X server also scales fonts. Hence it is important to set the font path in such a way, as to first have unscaled fonts, and then Type1 or TrueType fonts that are made to be scaled, and then finally bitmap fonts that will be scaled if needed. My font path is specified in the *XF86Config* file, and it looks like this:

```
    FontPath   "/usr/X11R6/lib/X11/fonts/misc/:unscaled"
    FontPath   "/usr/X11R6/lib/X11/fonts/75dpi/:unscaled"
    FontPath   "/usr/local/Office51/fonts/75dpi/:unscaled"
    FontPath   "/usr/local/lib/X11/fonts/75dpi/:unscaled"
    FontPath   "/usr/X11R6/lib/X11/fonts/Type1/"
    FontPath   "/usr/local/Office51/fonts/type1/"
    FontPath   "/usr/local/lib/X11/fonts/Type1/"
    FontPath   "/usr/X11R6/lib/X11/fonts/Speedo/"
    FontPath   "/usr/X11R6/lib/X11/fonts/PEX/"
    FontPath   "/usr/X11R6/lib/X11/fonts/misc/"
    FontPath   "/usr/X11R6/lib/X11/fonts/75dpi/"
    FontPath   "/usr/local/Office51/fonts/75dpi/"
    FontPath   "/usr/local/Corel/shlib10/fonts/"
#    FontPath   "/usr/X11R6/lib/X11/fonts/100dpi/:unscaled"
#    FontPath   "/usr/X11R6/lib/X11/fonts/100dpi/"
#    FontPath   "/usr/X11R6/lib/X11/fonts/greek/"
```

I am using a laptop, so the 75dpi fonts are fine for me. On a monitor the 100dpi fonts might look better. The order in the font path is important, X will grab the first match, not the best match.

Many documents specify nowadays the font face `arial`, which is usually not available on a Linux box. If you run a newer version of XFree (I think version > 3.98), then you have support for TrueType Fonts, so install them from your Windows partition into a directory X can read, and add it to the font path. Make sure you create a *fonts.dir* file.

I run an older version of X, so I decided to transform the arial and times TrueType fonts to something my X understands. There are at least two choices, I used the programs *ttf2bdf*, see Section , and *ttf2pt1*, see Section 102.1. The first of those two choices creates bitmapped fonts, and I used it to make those fonts that are often used at sizes that are often used and that I did not yet have. See Section for the details. For the Type1 fonts read on. I transformed the normal, bold, italic, and bold-italic fonts. I copied the TrueType fonts into the empty directory where I ultimately want the new X fonts, and ran:

```
for i in *.ttf; do ttf2pt1 -b $i; done
```

Now I removed the `*.ttf` files. The *fonts.dir* directory I made by hand.

fonts.dir

```
8
arial.pfb   -microsoft-arial-medium-r-normal--0-0-0-0-p-0-iso8859-1
arialbd.pfb -microsoft-arial-bold-r-normal--0-0-0-0-p-0-iso8859-1
arialbi.pfb -microsoft-arial-bold-i-normal--0-0-0-0-p-0-iso8859-1
ariali.pfb  -microsoft-arial-medium-i-normal--0-0-0-0-p-0-iso8859-1
times.pfb   -microsoft-times-medium-r-normal--0-0-0-0-p-0-iso8859-1
timesbi.pfb -microsoft-times-bold-i-normal--0-0-0-0-p-0-iso8859-1
timesbd.pfb -microsoft-times-bold-r-normal--0-0-0-0-p-0-iso8859-1
timesi.pfb  -microsoft-times-medium-i-normal--0-0-0-0-p-0-iso8859-1
```

fonts.alias

I also wanted the arial fonts in this directory to replace the scaled bitmap helvetica fonts, that is done by having them aliased. Also, Netscape tends to look for fonts of the adobe family, and not the microsoft family, so I aliased that as well.

```
-adobe-arial-medium-r-normal--0-0-0-0-p-0-iso8859-1 \
        -microsoft-arial-medium-r-normal--0-0-0-0-p-0-iso8859-1
-adobe-arial-bold-r-normal--0-0-0-0-p-0-iso8859-1 \
        -microsoft-arial-bold-r-normal--0-0-0-0-p-0-iso8859-1
-adobe-arial-bold-i-normal--0-0-0-0-p-0-iso8859-1 \
        -microsoft-arial-bold-i-normal--0-0-0-0-p-0-iso8859-1
-adobe-arial-medium-i-normal--0-0-0-0-p-0-iso8859-1 \
        -microsoft-arial-medium-i-normal--0-0-0-0-p-0-iso8859-1
-adobe-helvetica-medium-r-normal--0-0-0-0-p-0-iso8859-1 \
        -microsoft-arial-medium-r-normal--0-0-0-0-p-0-iso8859-1
-adobe-helvetica-bold-r-normal--0-0-0-0-p-0-iso8859-1 \
        -microsoft-arial-bold-r-normal--0-0-0-0-p-0-iso8859-1
-adobe-helvetica-bold-i-normal--0-0-0-0-p-0-iso8859-1 \
        -microsoft-arial-bold-i-normal--0-0-0-0-p-0-iso8859-1
-adobe-helvetica-medium-i-normal--0-0-0-0-p-0-iso8859-1 \
        -microsoft-arial-medium-i-normal--0-0-0-0-p-0-iso8859-1
Arial-MT           -microsoft-arial-medium-r-normal--0-0-0-0-p-0-iso8859-1
Arial-BoldItalicMT -microsoft-arial-bold-i-normal--0-0-0-0-p-0-iso8859-1
Arial-BoldMT       -microsoft-arial-bold-r-normal--0-0-0-0-p-0-iso8859-1
Arial-ItalicMT     -microsoft-arial-medium-i-normal--0-0-0-0-p-0-iso8859-1
-adobe-times-medium-r-normal--0-0-0-0-p-0-iso8859-1 \
        -microsoft-times-medium-r-normal--0-0-0-0-p-0-iso8859-1
-adobe-times-bold-i-normal--0-0-0-0-p-0-iso8859-1 \
        -microsoft-times-bold-i-normal--0-0-0-0-p-0-iso8859-1
-adobe-times-bold-r-normal--0-0-0-0-p-0-iso8859-1 \
        -microsoft-times-bold-r-normal--0-0-0-0-p-0-iso8859-1
-adobe-times-medium-i-normal--0-0-0-0-p-0-iso8859-1 \
        -microsoft-times-medium-i-normal--0-0-0-0-p-0-iso8859-1
TimesNewRomanPSMT \
        -microsoft-times-medium-r-normal--0-0-0-0-p-0-iso8859-1
TimesNewRomanPS-BoldMT \
        -microsoft-times-bold-i-normal--0-0-0-0-p-0-iso8859-1
TimesNewRomanPS-BoldItalicMT \
        -microsoft-times-bold-r-normal--0-0-0-0-p-0-iso8859-1
```

```
TimesNewRomanPS-ItalicMT \
        -microsoft-times-medium-i-normal--0-0-0-0-p-0-iso8859-1
```

Notice that the lines ending on a backslash should really be continued with what is printed in the line following it, it just did not fit above.

82.2 Highlighting and such

The *.Xdefaults* also allows for other modifications, they could also be written into the *Netscape* file in */var/X11R6/lib/app-defaults/*.

```
! These resources control the foreground and background colors of text
! which has been selected with the mouse. I chose Yellow and Blue.
!
Netscape*selectForeground:                          #FFFF00
Netscape*selectBackground:                          #0000FF

! Disable the four buttons "Netscape", "Shop", "Search", and "Security"
Netscape*toolBar.destinations.isEnabled: false
Netscape*toolBar.myshopping.isEnabled: false
Netscape*toolBar.search.isEnabled: false
Netscape*toolBar.viewSecurity.isEnabled: false

! Trick Netscape into accepting symbol fonts
Netscape*documentFonts.charset*adobe-fontspecific:   iso-8859-1

! Disable the initial display of the splash screen
Netscape*noAboutSplash: true
```

83 Netwide Assembler

Files:
*/usr/local/nasm/**

Sources obtainable at:
`ftp://ftp.us.kernel.org/pub/software/devel/nasm/source/`
`http://www.web-sites.co.uk/nasm/`

```
configure --prefix=/usr/local/nasm --disable-nls
make
mkdir -p /usr/local/nasm/bin /usr/local/nasm/man/man1
make install
strip /usr/local/nasm/bin/*
gzip -9 /usr/local/nasm/man/man1/*
```

84 Open Motif

Sources obtainable at:
`ftp://ftp.sgi.com/other/motifzone/`
`http://www.openmotif.net/download/`

For a similar package see LessTif (section 51.1.5).

I downloaded the pre-compiled runtime and development packages, and installed those into */tmp*. Then I got rid of all but the libraries, header files, and some documentation. I put the rest into a tar archive and used Slackware's pkgtool to install.

85 Patch

Files:
/usr/bin/patch
/usr/man/man1/patch.1.gz

Sources obtainable at:
`ftp://ftp/gnu.ai.mit.edu/pub/gnu/patch/`

The new version of `patch` compiles straight out of the box.

```
./configure --prefix=/usr --disable-nls
make CFLAGS="-O2 -s"
make install
gzip -9 /usr/man/man1/patch.1
```

86 Patches

Patches contain upgrades for source code. To create a patch of a single file copy the file to another name, say *file.old* and make the desired changes on *file* itself. Then run the context sensitive *diff* command and save the output to a patch file, like so: `diff -c file.old file > file.patch`. If later on you need to apply the patch to the unpatched version then run in the directory of the unpatched file the command `patch < file.patch`. The patch file can be stored anywhere on the file system, just specify the full path.

Patching multiple source files in a directory is similar. First copy the whole directory to say *dir.old*, then modify all files as desired. Then run `diff` recursively from the directory in which both *dir* and *dir.old* are located, like so: `diff -cr dir.old dir > dir.patch`. To later apply the patch to an unpatched copy of the sources run the command `patch -p0 < dir.patch` from the directory in which *dir* is located. The `-p0` flag tells `patch` to use the path information that is stored in the patch file.

At times you want to undo a patch. This is possible by applying the patch to a patched version of the source and specifying the `-R` flag.

87 Pbm Plus

Files:
/usr/local/bin/{atktopbm, brushtopbm, cmuwmtopbm, icontopbm, gemtopbm, macptopbm, mgrtopbm, pbmlife, pbmmake, pbmmask, pbmreduce, pbmtext, pbmto10x, pbmtoascii, pbmtoatk, pbmtobbnbg, pbmtocmuwm, pbmtoepson, pbmtogem, pbmtogo, pbmtoicon, pbmtolj, pbmtomacp, pbmtomgr, pbmtopi3, pbmtoplot, pbmtoptx, pbmtox10bm, pbmtoxbm, pbmtoybm, pbmtozinc, pbmupc, pi3topbm, xbmtopbm, ybmtopbm, fitstopgm, fstopgm, hipstopgm, lispmtopgm, pgmbentley, pgmenhance, pgmhist, pgmnorm, pgmoil, pgmramp, pgmtofits, pgmtofs, pgmtolispm, pgmtopbm, psidtopgm, rawtopgm, pgmcrater, pgmedge, pgmtexture, giftoppm, gouldtoppm, ilbmtoppm, imgtoppm, mtvtoppm, pcxtoppm, pgmtoppm, pi1toppm, picttoppm, pjtoppm, ppmdither, ppmhist, ppmmake, ppmquant, ppmrelief, ppmtoacad, ppmtogif, ppmtoicr, ppmtoilbm, ppmtopcx, ppmtopgm, ppmtopi1, ppmtopict, ppmtopj, ppmtopuzz, ppmtorgb3, ppmtosixel, ppmtotga, ppmtouil, ppmtoxpm, ppmtoyuv, qrttoppm, rawtoppm, rgb3toppm, sldtoppm, spctoppm, sputoppm, tgatoppm, ximtoppm, xpmtoppm, yuvtoppm, ppmforge, ppmpat, ppmquantall, pnmarith, pnmcat, pnmconvol, pnmcrop, pnmcut, pnmdepth, pnmenlarge, pnmfile, pnmflip, pnminvert, pnmnoraw, pnmpaste, pnmscale, pnmtile, pnmtops, pnmtorast, pnmtoxwd, rasttopnm, xwdtopnm, pnmgamma, pnmrotate, pnmshear, tifftopnm, pnmtotiff, anytopnm, pnmindex, pnmmargin, pnmsmooth, g3topbm.pbm, pbmtog3.pbm}
/usr/local/man/man{1,5} (The corresponding man pages.)

Unpack the sources and `cd pbmplus10Dec91`. Then type `xmkmf; make Makefiles`. After that `cd libtiff` and `cp Makefile.dist Makefile; chmod 644 Makefile`. Now this Makefile needs to be edited. Change line 65 to read

```
CFLAGS=       -O2 -m486 ${IPATH} ${CONF_LIBRARY}
```

Finally, `cd ..` and `make` compiles all the binaries. The commands `make install` and `make install.man` install all of the binaries and the man pages. I moved the binaries into */usr/local/bin* (they got installed into */usr/X11/bin/* by default), and I moved the man pages as well. As I installed a different version of *g3topbm* and of *pbmtog3* earlier that I intended to keep, I changed the corresponding names in the **pbmplus** package to *g3topbm.pbm* and to *pbmtog3.pbm*, respectively. Finally, I updated the *whatis* database of the man pages.

I finally uninstalled this package and replaced it with the **netpbm** package, because latex2html needs that later package.

88 PCMCIA and Cardbus Services

Files:
*/lib/modules/2.0.36/pcmcia/**
/lib/modules/2.0.36/net/8390.o
/sbin/{cardmgr, cardctl, ifport, ifuser, scsi_info, pcinitrd}
/sbin/{ftl_format, ftl_check}
/sbin/{dump_cis, pack_cis}
/usr/X11R6/bin/cardinfo
/usr/man/man4/{pcmcia_core.4.gz, tcic.4.gz, i82365.4.gz, tc589_cs.4.gz, pcnet_cs.4.gz,

dummy_cs.4.gz, serial_cs.4.gz, ftl_cs.4.gz, memory_cs.4.gz, fmvj18x_cs.4.gz, tc574_cs.4.gz, tc575_cb.4.gz, netwave_cs.4.gz, wavelan_cs.4.gz, aha152x_cs.4.gz, xirc2ps_cs.4.gz, memory_cb.4.gz, ibmtr_cs.4.gz}
/usr/man/man5/{pcmcia.5.gz, stab.5.gz}
/usr/man/man8/{cardmgr.8.gz, cardctl.8.gz, ftl_format.8.gz, ftl_check.8.gz, scsi_info.8.gz, ifport.8.gz, ifuser.8.gz, ide_info.8.gz, pcinitrd.8.gz, pack_cis.8.gz, dump_cis.8.gz}
/etc/rc.d/{rc.pcmcia, rc.pcmcia.N} */etc/conf.modules*
*/etc/pcmcia/**
/etc/rc.d/rc.S

Sources obtainable at:
`http://www.hyper.stanford.edu`

```
make config
make all
make install
```

Then compress all the man pages. Also, edit the file */etc/rc.d/rc.S* so that it contains the line `. /etc/rc.d/rc.pcmcia start`. Finally make sure that the kernel daemon is running to enable loading of the modules. The kernel daemon is also started from within the file */etc/rc.d/rc.S*.

The configure stage

```
# make config

Linux PCMCIA Configuration Script

The default responses for each question are correct for most users.
Consult the PCMCIA-HOWTO for additional info about each option.

Linux source directory [/usr/src/linux]:

The kernel source tree is version 2.0.36.
The current kernel build date is Thu Apr 22 13:04:32 1999.

Alternate target install directory []:
  Module install directory [/lib/modules/2.0.36]:
C compiler name [gcc]:
Linker name [ld]:
Compiler flags for debugging []:
Build 'trusting' versions of card utilities (y/n) [y]:
Include 32-bit (CardBus) card support (y/n) [y]:

The PCMCIA drivers need to be compiled to match the kernel they
will be used with, or some or all of the modules may fail to load.
If you are not sure what to do, please consult the PCMCIA-HOWTO.

How would you like to set kernel-specific options?
    1 - Read from the currently running kernel
    2 - Read from the Linux source tree
```

```
    3 - Set each option by hand (experts only!)
Enter option (1-3) [2]:

Kernel configuration options:
    Symmetric multiprocessing support is disabled.
    PCI BIOS support is enabled.
    Advanced Power Management (APM) support is enabled.
    SCSI support is enabled.
    Networking support is enabled.
     Radio network interface support is disabled.
     Token Ring device support is disabled.
    Module version checking is enabled.
    PCMCIA IDE device support is disabled.
    DEC Alpha UDB target platform is disabled.
    /proc filesystem support is enabled.

Your module utilities are version 2.0.0.

It looks like you have a BSD-ish init file setup.
    You'll need to edit /etc/rc.d/rc.S to invoke /etc/rc.d/rc.pcmcia
    so that PCMCIA services will start at boot time.

X Windows include files found.
/usr/X11/lib/libforms.so found.
```

89 Perl

Files:
/usr/bin/{a2p, h2ph, c2ph, pstruct, s2p, find2perl, perl, perlbug, perl5.00503, h2xs, perldoc, pod2html, pod2latex, pod2man, pod2text, pl2pm, splain, perlcc}
/usr/man/man1/{perlop.1.gz, perl.1.gz, perlre.1.gz, perldebug.1.gz, perlapio.1.gz, perlbook.1.gz, perlbot.1.gz, perlcall.1.gz, perldata.1.gz, perlembed.1.gz, perldiag.1.gz, perldelta.1.gz, perl5004delta.1.gz, perldsc.1.gz, perllocale.1.gz, perlfaq.1.gz, perlfaq1.1.gz, perlfaq2.1.gz, perlfaq3.1.gz, perlfaq4.1.gz, perlfaq5.1.gz, perlfaq6.1.gz, perlfaq7.1.gz, perlfaq8.1.gz, perlfaq9.1.gz, perlform.1.gz, perlfunc.1.gz, perlguts.1.gz, perlhist.1.gz, perlipc.1.gz, perlxstut.1.gz, perlxs.1.gz, perllol.1.gz, perlmod.1.gz, perlmodinstall.1.gz, perlmodlib.1.gz, perlobj.1.gz, perlpod.1.gz, perlopentut.1.gz, perlport.1.gz, perlref.1.gz, perlreftut.1.gz, perlrun.1.gz, perlsec.1.gz, perlstyle.1.gz, perlsub.1.gz, perlsyn.1.gz, perlthrtut.1.gz, perltie.1.gz, perltoc.1.gz, perltoot.1.gz, perltrap.1.gz, perlvar.1.gz, pl2pm.1.gz, c2ph.1.gz, h2ph.1.gz, h2xs.1.gz, perldoc.1.gz, perlbug.1.gz, splain.1.gz, s2p.1.gz, a2p.1.gz, pod2man.1.gz, pod2html.1.gz, pstruct.1.gz, xsubpp.1.gz}
*/usr/lib/perl5/**
*/usr/info/perl**

Sources obtainable at:
`http://www.perl.com`
`ftp://ftp.hkstar.com/pub/CPAN/src/`

Remove the old installation using `pkgtool` if you have the original slackware installation still around. Unpack the sources from */usr/src* and `cd perl-5.005_03`, or whatever the current verswion might be. Run the script `sh Configure`, which will ask a lot of questions. Go with the default for almost all. Change the compiler to `gcc`, get rid on *shlib* in the list of libraries, set the email address to `mayer@127.0.0.1`, Run `make depend` when asked. Then run `make` and `make test`, and finally `make install`. Compress all the man pages in */usr/local/man/man1* and in */usr/lib/perl5/5.00503/man/man3*. Add that last directory (without the "man3") to the MANPATH in */etc/profile*. Also, I once found some old info pages for perl, which I keep in */usr/info*. Finally, after the installation is done, and after you rebooted so that the new MANPATH is set, it might be a good idea to run `makewhatis -v -w` to update the whatis database for the `man -k` command.

If you want to install HTML formatted documentation, run the command

```
./installhtml --recurse --htmlroot=/usr/local/html/perl --verbose
cd /usr/local/html/perl
for i in html/*; do ln -s $i; done
```

The last command links the directories from one level up, because some documents cross-reference to that location.

90 Pretty Good Privacy (PGP)

Pretty Good Privacy

Files:
*~/.pgp/**
/usr/local/bin/pgp
*/usr/local/lib/pgp/**
/usr/local/man/cat1/pgp.1.gz

PGP, short for Pretty Good Privacy, is a public key encryption package; with it, you can secure messages you transmit against unauthorized reading and digitally sign them so that people receiving them can be sure they come from you. It is widely distributed on the Internet, the official version is at MIT. A complete mirror is at `ftp://ftp.funet.fi/pub/crypt/cryptography/pgp/`. Here is the guts of *setup.doc* that comes with the distribution.

You likely will have to compile PGP for your system; to do this, first make sure the unpacked files are in the correct unix text-file format (the files in *pgp262s.zip* are in MSDOS CRLF format, so for Unix you must unpack with `unzip -a`; the tar file *pgp262s.tar.gz* uses normal Unix line feed conventions).

You will need the RSAREF package written by RSA Data Security. It is included with the current PGP distribution from MIT. Use this version as it has been edited for the larger key sizes needed by PGP 2.6.2 as well as to improve performance.

When you untar *pgp262s.tar.gz* you will find that it contains 5 files. *pgp262si.tar* contains all non-binary files for PGP including all source code. This tar archive has been created assuming that you will untar it directly into your PGP 2.6.2 "build" directory. The file *pgp262si.asc* is a detached digital signature of *pgp262si.tar* (which you can verify after you have PGP operating—actually

not, because I didn't keep *pgp262si.tar*). *rsaref.tar* contains the source code for the RSAREF distribution. You should also untar it from your PGP "build" directory. All RSAREF software will automatically go into an *rsaref* subdirectory. *I have already done all these steps on my backup, no more zip files for me.* `unzip -a` didn't exactly work smoothly, I had to specify that the text files should be converted, but others should not because they contained compressed data.

After switching to ELF pgp did no longer compile. Here is the problem: Pgp uses two assembler routines which cause problems when compiling for ELF because ELF handles symbols differently than a.out. In ELF labels are simply called LABEL while in the latter they carry an underscore _LABEL. I modified the binaries to fix the labels and modified the makefile. All this is in *ELF_fix.tgz* on the floppy.

Then *cd rsaref/install/unix* and invoke the *make* command there to build the RSAREF software library. This will try to create an *rsaref.a* library in the current directory. After RSAREF is successfully built, *cd ../../../src* from the *rsaref/install/unix* directory and invoke `make linux-elf` there. If all goes well, you will end up with an executable file called *pgp*.

Before you install *pgp*, run these tests (do not create your real public key yet, this is just for testing pgp):

- create a public/secret key pair (enter `test` as userid/password): *pgp -kg*
- add the keys from the file *keys.asc* to the public keyring: *pgp -ka keys.asc*; *pgp* will ask if you want to sign the keys you are adding, answer yes for at least one key.
- do a keyring check: *pgp -kc*
- encrypt pgpdoc1.txt: *pgp -e pgpdoc1.txt test -o testfile.pgp*
- decrypt this file: *pgp testfile.pgp* this should produce the file *testfile*. Compare this file with *pgpdoc1.txt*.

If everything went well, install *pgp* into */usr/local/bin*.

Place the documentation, *pgpdoc1.txt* and *pgpdoc2.txt* somewhere where you can reasonably read it. I put it into */usr/local/lib/pgp*. The software looks for it when running (especially generating keys), so someplace reasonably obvious would be good. *pgp -kg* will give you full details if it can't find the manuals. I also copied almost all other documentation that came with it into this directory (*appnote.doc, keyserv.doc, pgp.hlp, politic.doc, setup.doc, changes.doc, language.txt, pgp262si.asc, readme.doc, config.txt, mitlicen.txt, pgpdoc1.txt, rsalicen.txt, keys.asc, pgformat.doc, pgpdoc2.txt, rsaref.asc*).

Format the man page with `nroff -man pgp.1 | gzip -9c > pgp.1.gz` and install it into the directory */usr/local/man/cat1*.

Create a subdirectory somewhere in your home directory hierarchy to hold your public and private key rings and anything else *pgp* might need (like the language.txt file). The default name PGP assumes is *~/.pgp*. If you want to use a different name, you must set the environment variable `PGPPATH` to point to this place before you use the system. Copy *config.txt* into this directory. **IMPORTANT:** This directory cannot be shared! It will contain your personal private keys!

If you are installing PGP system-wide, the directory to use is */usr/local/lib/pgp* for the config, language and help files. This can be changed in fileio.h when compiling. It's the value of `PGP_SYSTEM_DIR`.

Now, if you haven't done so yet, *go read the manual*. At least read the man page. Then certify a few of the keys in the public key ring.

Mkpgp

Files:
~/.pgp/mkpgp.rc
/usr/local/bin/mkpgp
/usr/local/man/man1/mkpgp.1.gz

This is a csh script that allows to use pgp from within pine. I edited the script so it takes *~/.pgp* as its default directory. The idea is that you use this program as an alternate editor for pine after you have written your message. That also means that you need to be in write mode to use it, so to decode a received message you need to start forwarding it, because that lets you edit the text. They just call the alternate editor from within pine, and it will do it's thing. All you need to do to set this up is to enable the alternate editor command from within pine.

It did not come with a man page, so I essentially installed the *README* that came with the distribution.

Mailcrypt

Files:
/usr/share/emacs/site-lisp/{mailcrypt.el,mailcrypt.elc,mc-pgp.el,mc-pgp.elc, mc-remail.el,mc-remail.elc,mc-toplev.el,mc-toplev.elc}
/usr/info/mailcrypt.info.gz

This adds minor modes to emacs to allow very convenient access to pgp from within emacs. Unpack the sources, and run `make` to byte-compile the lisp files. Then copy them to the site-lisp directory, and install the info page. This package needs to be loaded and configured from within the system or the personal emacs startup file. I edited the system startup file */usr/share/emacs/site-lisp/default.el*.

```
(autoload 'mc-install-write-mode "mailcrypt" nil t)
(autoload 'mc-install-read-mode "mailcrypt" nil t)
(add-hook 'mail-mode-hook 'mc-install-write-mode)
(add-hook 'rmail-mode-hook 'mc-install-read-mode)
(add-hook 'rmail-summary-mode-hook 'mc-install-read-mode)
(add-hook 'gnus-summary-mode-hook 'mc-install-read-mode)
(add-hook 'news-reply-mode-hook 'mc-install-write-mode)
```

91 Point-to-point Internet Connection (PPP)

Files:
/usr/sbin/{chat, pppd, pppstats}
/usr/man/man8/{pppd.8.gz, chat.8.gz, pppstats.8.gz}
*/etc/ppp/**

/usr/local/bin/{ppp-on, tppp-on, cppp-on, ppp-off}

/usr/src/linux/drivers/net/bsd_comp.c
/usr/src/linux/include/linux/{if_ppp.h, if_pppvar.h, ppp-comp.h, ppp_defs.h}
/usr/src/linux/drivers/net/ppp.c
/usr/src/linux/drivers/Makefile
/usr/include/net/{if_ppp.h, if_pppvar.h, ppp-comp.h, ppp_defs.h}
/lib/modules/2.2.12/net/ppp-compress-21.o

This describes *ppp* version 2.2.0f, obtainable from any linux site under *system/networks/ppp*. To run *ppp* the kernel needs to be configured to use it, too. Hence you might need to recompile the kernel. I upgraded to version 2.3.6 by now.

The compilation of *ppp* involves a few steps. First, `./configure` creates the right setup. Then `install kernel` installs a few header files and driver files into the kernel sources. Finally `make install` finishes off the installation stage.

Also install the scripts from the *scripts* subdirectory into */etc/ppp*. They will need editing. Then link those that need to be run by users to */usr/local/bin*. Furthermore, I want any user to be able to enable/disable PPP, and for this *ppp-off* needs to be SUID root. However, shell scripts cannot be SUID root under linux. So I wrote a C version of the shell script. It is also a bit more sophisticated, it checks whether the process ID from the process ID file actually corresponds to a running pppd process. If necessary, it removes a stale process ID file. Finally, the device file */dev/ttyS1* needs to be accessible for all users, so run `chmod 666 /dev/ttyS1`. I think this is really only so if one uses `chat` to dial up, because `pppd` runs SUID root anyways.

There is a compression module compiled, it is called *bsd_comp.o* and resides in the directory */lib/modules/2.2.12/net*, where 2.2.12 is of course the kernel version. However, when connecting to t-online, the ppp daemon expects the compression module to be called *ppp-compress-21.o*, so I created a symbolic link. For this module to be loaded correctly with `modprobe`, you need to run `depmod -a` first. This is done automatically at system boot.

You can also use *dip* to dial up PPP, which is what I used to do. Nowadays I use PAP secrets and chat to connect, see the setup files below.

pap-secrets

```
# Secrets for authentication using PAP
# client                        server        secret          IP addresses
# the first account
YourUsername                    *             YourPassword
# account at C-server
Your-C-UserName                 *             Your-C-Password
```

ppp-on

```
#!/bin/sh
cat << EOF

Please use one of the following:

   cppp-on to connect to C-server in Germany
   pppp-on to connect to account long distance
```

```
   lppp-on to connect to account long distance using predial

EOF
```

pppp-on

```
#!/bin/sh

localip=0.0.0.0
remoteip=
device=/dev/ttyS1

pppflags="115200 modem \
ipcp-accept-remote ipcp-accept-local \
defaultroute user YOURUSERNAME"

/usr/sbin/pppd lock \
connect '/usr/sbin/chat -f /etc/ppp/pppp.chat' \
$device $pppflags $localip:$remoteip
```

pppp.chat

```
TIMEOUT 180
ABORT "NO CARRIER"
ABORT BUSY
ABORT ERROR
"" +++ATZ
OK ATDTyourphonenumber
CONNECT ""
```

cppp-on

```
#!/bin/sh
localip=0.0.0.0
remoteip=
device=/dev/ttyS1
pppflags="115200 modem \
ipcp-accept-remote ipcp-accept-local \
defaultroute user YOUR-C-USERNAME"
/usr/sbin/pppd lock \
connect '/usr/sbin/chat -f /etc/ppp/cppp.chat' \
$device $pppflags $localip:$remoteip
```

cppp.chat

```
TIMEOUT 30
ABORT "NO CARRIER"
ABORT BUSY
ABORT ERROR
"" +++ATZ
OK ATDTyour-c-phonenumber
CONNECT ""
```

ppp-off.c

```
/* Written by Uwe F. Mayer <mayer@tux.org> April 19,1997      */
/* This program should kill a running PPP daemon that uses the interface  */
/* ppp0. No warranties of any kind. */

#include <string.h>
#include <stdio.h>
#include <unistd.h>
#include <sys/types.h>
#include <signal.h>

/* ---------------------------------------------------------------------------*/

int cleanup(pidfile, pidfilename)
     FILE *pidfile;
     char *pidfilename;
{
  if (fclose(pidfile)==EOF)
    {
      fprintf(stderr,"There has been a problem closing %s.\n",pidfilename);
    };
  if (unlink(pidfilename)){
    fprintf(stderr,"Could not remove the stale pid file %s.\n",pidfilename);
    return(2);
  }
  else{
    fprintf(stderr,"Removed the stale pid file %s.\n",pidfilename);
  }
  return(1);
}

/* ---------------------------------------------------------------------------*/

int main (argc, argv)
     int argc;
     char *argv[];
{
  char procfilename[256], buffer[256], pid_string[256];
  const char *pidfilename="/var/run/ppp0.pid";
  int pid, result;
  FILE *pidfile, *procfile;

  if ((pidfile=fopen(pidfilename,"r"))==NULL) {
    fprintf(stderr,"Error: Cannot open the pid file %s. PPP not active.\n",
            pidfilename);
    return(1);
  }
  if (! fscanf(pidfile,"%s",&pid_string)) {
    fprintf(stderr,"Error: Cannot read %s.\n",pidfilename);
    return(cleanup(pidfile,pidfilename));
  }
```

```
  strcpy(procfilename,"/proc/");
  strcat(procfilename,pid_string);
  strcat(procfilename,"/stat");
  result=(procfile=fopen(procfilename,"r"))!=NULL &&
         fscanf(procfile,"%d%s",&pid,buffer) &&
         (! strcmp(buffer,"(pppd)"));
  fclose(procfile);
  if (! result){
    fprintf(stderr,
            "Error: The lock file does not contain a pppd process number.\n");
    return(cleanup(pidfile,pidfilename));
  } else {
    /* this really is a running pppd process */
    if (kill((pid_t)pid,SIGINT)){
      /* Problem delivering the signal, try to kill the process and then  */
      /* clean up ourselves. There might remain a stale lock file for the */
      /* serial port / modem which got used.                              */
      kill((pid_t)pid,SIGKILL);
      return(cleanup(pidfile,pidfilename));
    }
  }
  return(0);
}
```

older files : ppp-dial

```
#!/bin/sh
#
# Script to initiate a ppp connection. This is the first part of the
# pair of scripts. This is not a secure pair of scripts as the codes
# are visible with the 'ps' command.  However, it is simple.
#
# These are the parameters. Change as needed.
if [ "XX$1" = "XX-q" ]; then
MODEMVOLUME="M0"
shift
else
MODEMVOLUME="M1"
fi
if [ ! "XX$1" = "XX" ]; then
TELEPHONE=$1
else
TELEPHONE=xxxxx89       # The telephone number for the connection
fi
echo trying $TELEPHONE
ACCOUNT=YourUserName    # The account name for logon (as in 'George Burns')
PASSWORD=YourPassWord   # The password for this account (and 'Gracie Allen')
LOCAL_IP=0.0.0.0        # Local IP address if known. Dynamic = 0.0.0.0
REMOTE_IP=0.0.0.0       # Remote IP address if desired. Normally 0.0.0.0
NETMASK=255.255.0.0     # The proper netmask if needed
#
# Export them so that they will be available at 'ppp-on-dialer' time.
```

```
export TELEPHONE ACCOUNT PASSWORD MODEMVOLUME
#
# This is the location of the script which dials the phone and logs
# in.  Please use the absolute file name as the $PATH variable is not
# used on the connect option.  (To do so on a 'root' account would be
# a security hole so don't ask.)
#
DIALER_SCRIPT=/etc/ppp/ppp-on-dialer
#
# Initiate the connection
#
# I put most of the common options on this command. Please, don't
# forget the 'lock' option or some programs such as mgetty will not
# work. The asyncmap and escape will permit the PPP link to work with
# a telnet or rlogin connection. You are welcome to make any changes
# as desired. Don't use the 'defaultroute' option if you currently
# have a default route to an ethernet gateway.
#
exec /usr/sbin/pppd modem crtscts /dev/ttyS1 76800\
        noipdefault netmask $NETMASK defaultroute connect $DIALER_SCRIPT
```

older files : ppp-on-dialer

```
#!/bin/sh
#
# This is part 2 of the ppp-on script. It will perform the connection
# protocol for the desired connection.
#
#INITIALIZE='AT&C1&Q5W1L0'$MODEMVOLUME'S0=0S36=7S46=138S48=7'
INITIALIZE='AT'$MODEMVOLUME
exec chat                                                   \
        TIMEOUT            3                                \
        ABORT              '\nBUSY\r'                       \
        ABORT              '\nNO ANSWER\r'                  \
        ABORT              '\nRINGING\r\n\r\nRINGING\r'     \
        ''                 '\rAT'                           \
        'OK-+++\c-OK'      ATH0                             \
        TIMEOUT            30                               \
        OK                 $INITIALIZE                      \
        OK                 ATDT$TELEPHONE                   \
        CONNECT            '\d\r\r'                         \
        'serid:'--'serid:' $ACCOUNT                         \
        'Password?'        $PASSWORD                        \
        'succeeds.'        '\d\r\r\r'                       \
        TIMEOUT            1                                \
        'ccess'--'ccess'--'ccess'--'ccess'   ''             \
        TIMEOUT            30                               \
        'witch_'           ''                               \
        '>'                ppp
```

older files : ppp-on-University

```
#!/bin/sh
#
# Script to initiate a ppp connection. This is the first part of the
# pair of scripts. This is not a secure pair of scripts as the codes
# are visible with the 'ps' command.  However, it is simple.
#
if [ "XX$1" = "XX-q" ]; then
MODEMVOLUME="M0"
shift
else
MODEMVOLUME="M1"
fi
#
# Export so that they will be available at 'ppp-on-dialer' time.
export MODEMVOLUME
#
# This is the location of the script which dials the phone and logs
# in.  Please use the absolute file name as the $PATH variable is not
# used on the connect option.  (To do so on a 'root' account would be
# a security hole so don't ask.)
#
DIALER_SCRIPT=/etc/ppp/redialer
#
# Initiate the connection
#
# I put most of the common options on this command. Please, don't
# forget the 'lock' option or some programs such as mgetty will not
# work. The asyncmap and escape will permit the PPP link to work with
# a telnet or rlogin connection. You are welcome to make any changes
# as desired. Don't use the 'defaultroute' option if you currently
# have a default route to an ethernet gateway.
#
NETMASK=255.255.0.0        # The proper netmask if needed
#
exec /usr/sbin/pppd modem crtscts /dev/ttyS1 76800\
        noipdefault netmask $NETMASK defaultroute connect $DIALER_SCRIPT
```

older files : redialer

```
#!/bin/sh
######################################################################
#
# These parameters control the attack dialing sequence.
#
# Maximum number of attempts to reach the telephone number(s)
MAX_ATTEMPTS=10

# Delay between each of the attempts. This is a parameter to sleep
# so use "15s" for 15 seconds, "1m" for 1 minute, etc.
SLEEP_DELAY=1s
```

```
########################################################################
#
# This is a list of telephone numbers. Add new numbers if you wish
# and see the function 'callall' below for the dial process.
PHONE1=xxxxx89
PHONE2=xxxxx91
#PHONE3=xxxyy53
#PHONE4=xxxyy50
#PHONE5=xxxyy49
#PHONE6=xxxyy48
#PHONE7=xxxyy60
#PHONE8=xxxyy61

########################################################################
#
# If you use the ppp-on script, then these are passed to this routine
# automatically. There is no need to define them here. If not, then
# you will need to set the values.
#
ACCOUNT=YourUserName
PASSWORD=YourPassWord
#INITIALIZE='AT&C1&Q5W1L0'$MODEMVOLUME'S0=0S36=7S46=138S48=7'
INITIALIZE='AT'$MODEMVOLUME

########################################################################
#
# Function to initialize the modem and ensure that it is in command
# state. This may not be needed, but it doesn't hurt.
#
function initialize
{
    chat TIMEOUT 3 '' AT 'OK-+++\c-OK' $INITIALIZE OK
    return
}

########################################################################
#
# Script to dial a telephone
#
function callnumber
{
chat                                                            \
        ABORT                   '\nBUSY\r'                      \
        ABORT                   '\nNO ANSWER\r'                 \
        ABORT                   '\nRINGING\r\n\r\nRINGING\r'    \
        ''                      ATDT$1                          \
        CONNECT                 '\d\r\r'                        \
        'serid:'--'serid:'      $ACCOUNT                        \
        'Password?'             $PASSWORD                       \
        'succeeds.'             '\d\r\r\r'                      \
        TIMEOUT                 1                               \
```

```
        'ccess'--'ccess'--'ccess'--'ccess'      ''              \
        TIMEOUT              30                                 \
        'witch_'             ''                                 \
        '>'                  ppp
#
# If the connection was successful then end the whole script with a
# success.
#
    if [ "$?" = "0" ]; then
       exit 0
    fi

    return
}

#######################################################################
#
# Script to dial any telephone number
#
function callall
{
#   echo "dialing attempt number: $1" >/dev/console
    callnumber $PHONE1
    callnumber $PHONE2
#    callnumber $PHONE3
#    callnumber $PHONE4
#    callnumber $PHONE5
#    callnumber $PHONE6
#    callnumber $PHONE7
#    callnumber $PHONE8
#    callnumber $PHONE9
}

#######################################################################
#
# Initialize the modem to ensure that it is in the command state
#
initialize
if [ ! "$?" = "0" ]; then
   exit 1
fi

#
# Dial telephone numbers until one answers
#
attempt=0
while : ; do
    attempt=‘expr $attempt + 1‘
    callall $attempt
    if [ "$attempt" = "$MAX_ATTEMPTS" ]; then
        exit 1
    fi
```

```
    sleep "$SLEEP_DELAY"
done
```

92 Printable

Files:
/usr/local/bin/printable

The perl script `printable` filters the input and only allows newlines, tabs, and any character from a blank to a tilde to pass, using ASCII ordering. That includes essentially all printable 7-bit characters (carriage returns and formfeeds are excluded).

printable

```
#!/usr/bin/perl
#
# Written by Uwe F. Mayer <mayer@tux.org>
#
# This script filters the input stream so that only printable characters
# appear on the output. A final newline will be added if the input does not
# end with a one.
#
while ($line=<STDIN>) {
    $line=~s/[^\t -~]//g;  #exclude everything but tabs and space thru to tilde
    print("$line\n");
}
if ($line eq "0") {
    print("0\n");
}
```

93 Printer

Files:
/etc/hosts.equiv
/usr/local/lib/ghostscript/{lprsetup.sh, unix-lpr.sh}
/usr/local/lib/ghostscript/filt/{direct, indirect, gsif, bj200}
/usr/local/src/lpf.c
/usr/local/bin/lpf
/etc/printcap

First of all, I suggest reading the Printing-HOWTO. Roughly, printing works like this. A print command like `lpr -Pbj200 textfile` copies the file into the spool area */var/spool/lp1* and creates a companion file with the necessary print job information. The `lpd` takes over and sends the file to the printer using the information provided in */etc/printcap*. This file is only read when `lpd` is started. If you change it you need to restart the printer daemon `lpd`. **Warning:** If the printer is

off then your print job goes to the bit bucket and does not even create an error message. You'll have to resubmit the job.

The printer daemon should be started at boot time. This happens when */etc/rc.d/rc.inet2* is run. Here is the relevant part of that file.

```
NET="/usr/sbin"
IN_SERV="lpd"
# Start the various INET servers.
# For us this is only the printer daemon lpd
for server in ${IN_SERV}
do
 if [ -f ${NET}/${server} ]
 then
  echo -n " ${server} "
  ${NET}/${server}
 fi
done
```

Also, make sure your local host is listed in */etc/hosts.equiv*, that is it needs the two listings `localhost` and `tosca`.

The necessary files

```
lrwxrwxrwx  1  root  lp     13  Feb 14  1995  /usr/bin/lpc -> /usr/sbin/lpc*
-rws--s--x  1  root  lp  17412  Feb 14  1995  /usr/bin/lpq*
-rwx--s--x  1  root  lp  17412  Feb 14  1995  /usr/bin/lpr*
-rwx--s--x  1  root  lp  17412  Feb 14  1995  /usr/bin/lprm*
-rwxr-s---  1  root  lp  21508  Feb 14  1995  /usr/sbin/lpc
-rwxr-s---  1  root  lp  37892  Feb 14  1995  /usr/sbin/lpd
drwxrwxr-x  2  root  lp   1024  Feb 3   12:00 /var/spool/lp1/
-rw-rw-r--  1  root  lp      4  Feb 3   12:00 /var/spool/lp1/.seq
-rw-rw-r--  1  root  lp      0  Feb 3   12:00 /var/spool/lp1/acct
-rw-rw-r--  1  root  lp      0  Feb 3   12:00 /var/spool/lp1/errs
-rw-rw-r--  1  root  lp      0  Feb 3   12:00 /var/spool/lp1/lock
-rw-rw-r--  1  root  lp      0  Feb 3   12:00 /var/spool/lp1/logfile
-rw-rw-r--  1  root  lp      0  Feb 3   12:00 /var/spool/lp1/status
drwxrwxr-x  2  root  lp   1024  Feb 3   12:00 /var/spool/lpd/
-rw-rw-r--  1  root  lp      4  Feb 3   12:00 /var/spool/lpd/.seq
-rw-rw-r--  1  root  lp      0  Feb 3   12:00 /var/spool/lpd/lock
-rw-rw-r--  1  root  lp      0  Feb 3   12:00 /var/spool/lpd/lpd.lock
-rw-rw-r--  1  root  lp      0  Feb 3   12:00 /var/spool/lpd/status
```

The staircase effect

The filter *lpf* translates the unix newline command into the DOS linefeed / carriage return combination. If you don't do it you get the so-called staircase effect. Also, my printer skips lines at the end of the page, so I inserted newlines. The C-source is listed below. This needs to be compiled, and the object code moved into */usr/local/bin/lpf*.

```
#include <stdio.h>
#include <string.h>
#include <ctype.h>

#define FALSE 0
#define TRUE 1

#define FF 0x0C
#define CR 0x0D

int main()
{
  int c,line,begin,column,last_was_ff;

  line=0;
  column=0;
  begin=TRUE; /* is TRUE if it is the begin of a line */
  last_was_ff=FALSE;

  while ((c=getc(stdin)) != EOF)
    {
      if ((begin==TRUE) && (line%60==0)) {
if ((line == 60) && (last_was_ff == TRUE)) {
  /* after a form feed we need to print 61 lines, as the first line
     is the newline character after the form feed*/
  line=59;
  last_was_ff=FALSE;
}
else {
  if (line > 0) {
    /* finish off the current page */
    putc('\n',stdout);
    putc('\n',stdout);
    putc('\n',stdout);
  }
  /* start a new page */
  if (last_was_ff == FALSE)
    putc('\n',stdout);
  putc('\n',stdout);
  putc('\n',stdout);
}
      }
      begin=FALSE;
      putc(c,stdout);
      switch (c) {
      case '\n':
putc(CR,stdout); /* this is a DOS printer */
line++;
        column=0;
        begin=TRUE;
break;
      case FF:
```

```
        /* reset the program */
last_was_ff=TRUE;
line=0;
        column=0;
        begin=TRUE;
break;
      default:
        if (column++ == 80) {
          begin=TRUE;
          column=0;
          line++;
        }
      }
    }
  putc(FF,stdout);
  return(0);
}
```

The printcap entry

The listing below is for the text printer. For the postscript driver see Section 40.3. Also make also sure to comment-out the generic printer.

```
generic:\
  :lp=/dev/lp1:\
  :if=/usr/local/bin/lpf_generic:\
  :sd=/var/spool/lp1:\
  :lf=/var/spool/lp1/errs:\
  :mx#0:\
  :sh:\
  :sf:
# Canon BJ-200 ex bubble jet printer
lptext|text|bj200text:\
  :lp=/dev/lp1:\
  :if=/usr/local/bin/lpf:\
  :sd=/var/spool/lp1:\
  :lf=/var/spool/lp1/errs:\
  :mx#0:\
  :sh:\
  :sf:
```

94 Profile and Bashrc

Files:
/etc/profile
~/.profile
~/.bashrc

The */etc/profile* file is a system wide initialization script which is run at login time for each user, while *.profile* is the users own login initialization. The *.bashrc* file is an initialization file run

by each interactive invocation of the bash shell. My *.bashrc* simply sources the other two, so that all interactive shells get the same initialization. Aliases common to all users rest in */etc/profile*, others go into *.profile*.

95 Pstotext

Files:
/usr/local/bin/pstotext
/usr/local/man/man1/pstotext.1.gz

Sources obtainable at:
`http://www.research.digital.com/SRC/virtualpaper/pstotext.html`

The address to this utility was found in the Ghostscript HTML-formatted documentation. Compile with `make CC="gcc -O3"` and install by hand.

96 Qt

Files:
/usr/local/qt
/etc/profile
/etc/ld.so.conf

Sources obtainable at:
`http://www.troll.no`

This is a Graphical User Interface package ported to most UNIX/MS-Windows systems. **Warning:** Qt version 2.0.1 does not work with kde version 1.1.2, so I installed qt version 1.44 instead. The instructions below still mostly apply.

Unpack the archive from */usr/local* and rename *qt-version-number* to *qt*. Add the following to *etc/profile*.

```
QTDIR=/usr/local/qt
MANPATH=${MANPATH}:/usr/local/qt/man
export QTDIR
```

Add the following to */etc/ld.so.conf*.

```
/usr/local/qt/lib
```

For version 2.0.1, install with the system-wide installation of libz and libpng. For version 1.44 you enable GIF support by editing *include/gif_wt.h* and set the `#define` to equal 1. For version 2.3.0 you enable GIF support by editing *src/kernel/qgif.h* and set `QT_BUILTIN_GIF_READER` equal 1.

Hence from */usr/local/qt* run:

```
. /etc/profile
# for qt-2.3.0 type:
./configure -gif -thread -system-zlib -system-libpng -system-jpeg
# for qt-2.0.1 type:
```

```
# ./configure -gif -sm -system-zlib -system-libpng
# for qt-1.44 type:
# make linux-g++-shared
make
ldconfig -v
for i in /usr/local/qt/bin/*; do ln -s $i /usr/local/bin ; done
# for qt-2.0.1 type
chmod -R root.root /usr/local/qt
```

This is officially it. However, it leaves you with a 44 MB installation. I chose to get rid of the sources, and a few other things, and I am down to 15 MB.

```
cd /usr/local/qt
# remove all top-level files, but not the directories
rm -f *
# remove all source files with the exception of the header files
find src extensions \! -name \*.h \( -type f -o  -type l \) | xargs rm
# Another solution is to replace the symbolic links in the include
# directory with the files they are pointing to.
# remove some empty directories
find src extensions -type d | sort -r | xargs rmdir
# compress the man pages
gzip -9 man/man1/*.1 man/man3/*.3qt
# clean the tutorial directory
cd tutorial; make clean; cd ..
# for version 1.44 remove some of the extra documentation
rm -rf doc configs
```

Now, if you like any of the example binaries, then move them to the *bin* subdirectory. I kept the following: *dclock*, *dirview*, *life*, *movies*, *tetrix*, *tictac*. Once you have done that, remove the *examples* directory and strip the binaries. Don't forget to link them from */usr/local/bin*.

97 Realaudio Player

Files:
/usr/local/bin/rvplayer
*/usr/local/rvplayer5.0/**

/usr/local/bin/realplay
*/usr/local/RealPlayerG2/**

/usr/bin/X11/{realplay rpnphelper}
*/usr/lib/RealPlayer7/**
/usr/local/netscape/plugins/{raclass.zip rpnp.so}
/usr/share/applnk/RealPlayer.kdelnk
/usr/share/mimelnk/audio/{RealMedia.kdelnk RealMedia2.kdelnk SMIL.kdelnk}

Sources obtainable at:
`http://www.i2k.com/~jeffd/rpopen/` for the patch for version 5.0
`http://www.realaudio.com/` for the sources

Version 5.0

Unpack the sources from */usr/local*, `tar` creates a subdirectory named *rvplayer5.0* for the files. Change to that directory and strip the library files and executable with `strip *`. Fix ownership and permissions, and then place a wrapper script into */usr/local/bin* called *rvplayer*.

```
#!/bin/sh
export LD_LIBRARY_PATH=/usr/local/rvplayer5.0
export LD_PRELOAD=${LD_LIBRARY_PATH}/open.so
exec ${LD_LIBRARY_PATH}/rvplayer $*
```

Now it so happens, that with the newer versions of Linux, there is a problem. There is also a fix, but the fix doesn't quite work so well. Anyways, in the documentation of the fix it says the following:

```
The patch only does one specific thing: fixes a sound device write
error problem.  The symptom of this problem is when trying to play
audio, there is a short burst of sound and then the RealPlayer program
stops playing with a sound write error. The patch-tar-ball contains a
libc shared object that can be preloaded when running RealPlayer to
eliminate the sound device write error problem.  It also contains
source, complete with GPL and a Makefile if you need to recompile it.
An example script is supplied to show how to use it.  If you need to
know how it works, it's not exactly complicated, it just ignores the
NONBLOCK flag when accessing /dev/dsp.
```

So here it goes. Unpack the patchfile, and compile with `make`, and then put the file *open.so* into */usr/local/rvplayer5.0*. Then preload this shared object before you start *rvplayer*, this is done within the script above.

RealPlayer G2 alpha

This software is currently in alpha stage, but a binary is available that will run until March 2000. It needs the GNU libc library version 2.1 or higher. If you run the downloaded setup binary as root, it will automatically install into */usr/local/RealPlayerG2*. Just link the binary *realplay* from */usr/local/bin*.

Files:

*/usr/lib/RealPlayer7/**
/usr/lib/netscape/plugins/{*raclass.zip rpnp.so*}
/root/{*.mime.types .mailcap*}

RealPlayer Version 7

Comes as a setup-type binary. Run it as root, it installs into */usr/local/RealPlayer7*. Ownership needs to be fixed. The setup also installs a lot of files into the *.kde* and the *.gnome* subdirectories, the details are available in the file /root/mimeinstall.results. I deleted them all and only kept the changes to *.mime.types* and *.mailcap*. The Netscape plugin gets installed into */root/.netscape/plugins*, I moved the two files *rpnp.so* and *raclass.zip* to */usr/lib/netscape/plugins*. It also installs a link to */usr/local/RealPlayer7/rpnphelper* into a directory one chooses at installation time, I chose */usr/local/bin*. I then also linked *realplay* from there.

98 Rc

Files:
/etc/rc.d/{rc.S, rc.local, rc.serial, rc.inet1, rc.inet2}
/etc/inittab

inittab

This is the first configuration that is looked at at bootup. It specifies the run level and what *rc* files are run. It also specifies the number of virtual terminals that are accessed with the `Alt+Fn` keys. I chose to comment three of them out, because they just take up memory but I never use them.

```
#
# inittab This file describes how the INIT process should set up
# the system in a certain run-level.
#
# Version: @(#)inittab 2.04 17/05/93 MvS
#                                      2.10    02/10/95        PV
#                                      3.00    02/06/1999      PV
#
# Author: Miquel van Smoorenburg, <miquels@drinkel.nl.mugnet.org>
# Modified by: Patrick J. Volkerding, <volkerdi@ftp.cdrom.com>
#

# These are the default runlevels in Slackware:
#   0 = halt
#   1 = single user mode
#   2 = unused (but configured the same as runlevel 3)
#   3 = multiuser mode (default Slackware runlevel)
#   4 = X11 with KDM/GDM/XDM (session managers)
#   5 = unused (but configured the same as runlevel 3)
#   6 = reboot

# Default runlevel. (Do not set to 0 or 6)
id:3:initdefault:

# System initialization (runs when system boots).
si:S:sysinit:/etc/rc.d/rc.S
```

```
# Script to run when going single user (runlevel 1).
su:1S:wait:/etc/rc.d/rc.K

# Script to run when going multi user.
rc:2345:wait:/etc/rc.d/rc.M

# What to do at the "Three Finger Salute".
ca::ctrlaltdel:/sbin/shutdown -t5 -rf now

# Runlevel 0 halts the system.
l0:0:wait:/etc/rc.d/rc.0

# Runlevel 6 reboots the system.
l6:6:wait:/etc/rc.d/rc.6

# What to do when power fails (shutdown to single user).
pf::powerfail:/sbin/shutdown -f +5 "THE POWER IS FAILING"

# If power is back before shutdown, cancel the running shutdown.
pg:0123456:powerokwait:/sbin/shutdown -c "THE POWER IS BACK"

# If power comes back in single user mode, return to multi user mode.
ps:S:powerokwait:/sbin/init 3

# The getties in multi user mode on consoles an serial lines.
#
# NOTE NOTE NOTE adjust this to your getty or you will not be
#                able to login !!
#
# Note: for 'agetty' you use linespeed, line.
#       for 'getty_ps' you use line, linespeed and also use 'gettydefs'
#       for 'mgetty' you use <options> line
c1:1235:respawn:/sbin/agetty 38400 tty1 linux
c2:1235:respawn:/sbin/agetty 38400 tty2 linux
c3:1235:respawn:/sbin/agetty 38400 tty3 linux
#c4:1235:respawn:/sbin/agetty 38400 tty4 linux
#c5:1235:respawn:/sbin/agetty 38400 tty5 linux
#c6:12345:respawn:/sbin/agetty 38400 tty6 linux
# three terminals are enough for me

# Serial lines
#s1:12345:respawn:/sbin/agetty 19200 ttyS0 vt100
#s2:12345:respawn:/sbin/agetty 19200 ttyS1 vt100

# Dialup lines
#d1:12345:respawn:/sbin/agetty -mt60 38400,19200,9600,2400,1200 ttyS0 vt100
#d2:12345:respawn:/sbin/agetty -mt60 38400,19200,9600,2400,1200 ttyS1 vt100
# The next is for FAX and terminal access
#d3:12345:respawn:/usr/local/sbin/mgetty -x 2 ttyS1
```

```
# Runlevel 4 used to be for an X window only system, until we discovered
# that it throws init into a loop that keeps your load avg at least 1 all
# the time. Thus, there is now one getty opened on tty6. Hopefully no one
# will notice. ;^)
# It might not be bad to have one text console anyway, in case something
# happens to X.
x1:4:wait:/etc/rc.d/rc.4

# End of /etc/inittab
```

rc.S

This is the first *rc* file run at bootup. I changed the mounting of the file systems listed in */etc/fstab*, see Section . Here is the relevant part.

```
# mount file systems in fstab (and create an entry for /)
# but not NFS because TCP/IP is not yet configured
# Also no CD or MS-DOS/Windoze partitions
/sbin/mount -a -v -t nonfs,nomsdos,noiso9660,novfat
```

rc.0 and rc.6

These are actually the same file, *rc.0* is a symbolic link to *rc.6*.

```
#! /bin/sh
#
# rc.6 This file is executed by init when it goes into runlevel
# 0 (halt) or runlevel 6 (reboot). It kills all processes,
# unmounts file systems and then either halts or reboots.
#
# Version: @(#)/etc/rc.d/rc.6 1.50 1994-01-15
#
# Author: Miquel van Smoorenburg <miquels@drinkel.nl.mugnet.org>
# Modified by:  Patrick J. Volkerding, <volkerdi@ftp.cdrom.com>
#

  # Set the path.
  PATH=/sbin:/etc:/bin:/usr/bin

  # If there are SystemV init scripts for this runlevel, run them.
  if [ -x /etc/rc.d/rc.sysvinit ]; then
    . /etc/rc.d/rc.sysvinit
  fi

  # Set linefeed mode to avoid staircase effect.
  stty onlcr

  echo "Running shutdown script $0:"

  # Find out how we were called.
  case "$0" in
```

```
*0)
command="halt"
;;
*6)
command=reboot
;;
*)
echo "$0: call me as \"rc.0\" or \"rc.6\" please!"
exit 1
;;
  esac

  # Save the system time to the hardware clock using hwclock
  if [ -x /sbin/hwclock ]; then
    echo "Saving the system time to the hardware clock..."
    if [ -e /etc/hardwareclock ]; then
      HC=‘cat /etc/hardwareclock | while read LINE ; do
        if [ "$LINE" = "UTC" -o "$LINE" = "utc" ]; then
          echo utc
          break;
        elif [ "$LINE" = "localtime" ]; then
          echo localtime
          break;
        fi
      done‘
    else
      # Default to localtime if there is no /etc/hardwareclock:
      HC=localtime
    fi
    # /sbin/hwclock --$HC --systohc
    # The line above *should* work with --localtime, but doesn’t.  Omitting
    # the argument defaults it to localtime, so use that.
    if [ "$HC" = "utc" ]; then
      /sbin/hwclock --utc --systohc
    else
      /sbin/hwclock --systohc
    fi
  fi

  # Kill all processes.
  # INIT is supposed to handle this entirely now, but this didn’t always
  # work correctly without this second pass at killing off the processes.
  # Since INIT already notified the user that processes were being killed,
  # we’ll avoid echoing this info this time around.
  if [ "$1" != "fast" ]; then # shutdown did not already kill all processes
    killall5 -15
    sleep 5
    killall5 -9
  fi

  # Try to turn off quota:
  if fgrep quota /etc/fstab 1> /dev/null 2> /dev/null ; then
```

```
    if [ -x /sbin/quotaoff ]; then
      echo "Turning off filesystem quotas."
      /sbin/quotaoff -a
    fi
  fi

  # Turn off accounting:
  if [ -x /sbin/accton ]; then
echo "Turning off accounting."
/sbin/accton
  fi

  # Before unmounting file systems write a reboot or halt record to wtmp.
  $command -w

  # Carry a random seed between reboots.
  # Save 512 bytes, which is the size of the entropy pool.
  echo "Saving random seed from /dev/urandom in /etc/random-seed..."
  dd if=/dev/urandom of=/etc/random-seed count=1 bs=512 2> /dev/null

  # Unmount any remote filesystems:
  echo "Unmounting remote filesystems."
  umount -a -tnfs

  # Turn off swap, then unmount local file systems.
  echo "Turning off swap."
  swapoff -a
  echo "Unmounting local file systems."
  # Don't remount UMSDOS root volumes:
  if [ ! "`mount | head -1 | cut -d ' ' -f 5`" = "umsdos" ]; then
    umount -a -tnonfs
    echo "Remounting root filesystem read-only."
    mount -n -o remount,ro /
  else
    umount -a -tnonfs -tnoumsdos
  fi
  # This never hurts:
  sync

  # Shut down RAID devices:
  if [ -r /etc/mdtab ]; then
    echo "Stopping RAID devices."
    /sbin/mdstop -a
  fi

  # See if this is a powerfail situation.
  if [ -f /etc/power_is_failing ]; then
    echo "Turning off UPS, bye."
    /sbin/powerd -q
    exit 1
  fi
```

```
  # Now halt (poweroff with APM kernels) or reboot.
  if [ "$command" = "reboot" ]; then
    echo "Rebooting."
    reboot -f
  else
    halt -f -p
  fi
```

rc.local

```
#! /bin/sh
# Put any local setup commands in here
# Running the general purpose mouse gpm
echo "Running gpm..."
gpm -t msc -3 &
```

rc.serial

This initializes the serial ports. I changed the following lines. Read the page `man setserial`. The hardware handshake thing I got from the Linux Networks Administrator's Guide.

```
${SETSERIAL} /dev/cua3 ${AUTO_IRQ} autoconfig spd_vhi
stty crtscts < /dev/cua3  #switches on hardware handshake on my modem
```

rc.inet1, rc.inet2

The first of the two sets local internet addresses, and the second one starts a few daemons. Read Sections 76.4 and 82.2.

99 GNU Readline Library

Files:
/usr/local/lib/libreadline.a
*/usr/local/include/readline/**
/usr/local/info/{readline.info.gz, history.info.gz}
/usr/local/man/man3/readline.3.gz

Sources obtainable at:
`http://www.ncftp.com/dowload`

`./configure` and `make` as usual. However, the *Makefile* does contain a typo, which will cause an error in the `make install` stage. This can be ignored. Now go to the *doc* subdirectory, and compress the files *readline.info, history.info, and readline.3*. Then install the man page and the info pages manually. Remove the file */usr/local/lib/libreadline.old*.

This library also comes as part of the bash distribution. Make bash, and as a side effect you will also make this library. If you want to install this way, then run `make -n install` in the *lib/readline* subdirectory to see what it wants to install, and then copy to the place where you want it.

100 Rescue Disk

The general idea is to compute a kernel and put it on a floppy. Then prepare a complete file system, compress an image of it, and put it onto the same floppy. Finally tell the kernel on the floppy where to find this compressed image and also tell it to load it into a ram-disk. The kernel compilation goes as usual, just create a kernel with everything that you need already in it, no modules. In particular it should have support for ftape, ram-disks (including initialization at boot), aout and elf binaries. Compile and copy to the floppy as in

```
dd if=/usr/src/linux/arch/i386/boot/zImage of=/dev/fd0 bs=1024
```

Make note of the number of blocks written, I'll assume here that it is less than 500.

Then get hold of a root disk (for example the slackware root disk *color*) and copy it on a file system of the desired size (4MB).

```
insmod rd                          # load the ramdisk driver if it's a module
zcat color.gz > /dev/ram1
mount -t ext2 /dev/ram1 /mnt1      # now /mnt1 has a complete root system
  # now prepare the second ram disk, first write zeros as the compress well
dd if=/dev/zero of=/dev/ram2 bs=1024 count=4096
  # and create the file system, 0% reserved for the super user
mke2fs -vm0 /dev/ram2 4096
  # now mount it and copy the root system into it
mount -t ext2 /dev/ram2 /mnt2
cd /mnt1
find . -print | cpio -pdv /mnt2
```

At this stage we have a complete file system on */mnt2*. However, it might not have all the binaries you want it to have. Note, all binaries you want to install must be aout binaries, as this is the only libc libraries that's going to be on the floppy. Let's deal with the libraries now.

```
cd \lib
cp ld.so /mnt2/lib
cp ld-linux.so.1.9.5 /mnt2/lib     # or whatever you have that
                                   # supports both elf and aout
cp libc.so.4.7.5 /mnt2/lib         # the last aout libc
cd /mnt2/lib
rm lclite.4.6.27
ln -sf libc.4.7.5 libc.so.4
ln -s /mnt/lib/libc.so.5           # so we can use elf binaries after
                                   # we mount the hard drive under /mnt
ln -s /mnt/lib/libc.so.6
ln -s /mnt/lib/libm.so.4
ln -s /mnt/lib/libm.so.5
ln -s /mnt/lib/libm.so.6
```

Now also copy all the binaries into */mnt2/* that you want on the rescue floppy. However, **they must be aout versions!** You most likely will have to compile them and cannot just use the ones from your hard drive. I installed *ps, free, mt, vi, cpio, df, e2fsck, hostname, lilo, gpm*, and as the upgrade of the C library caused problems with logging in also *login*.

Here are a few hints on how to recompile those binaries.

mt and **cpio** The slackware source, for example, is *source/a/cpio/cpio-2.3.tar.gz*. Unpack it, cd into the directory, and run

```
CC="gcc -V2.7.2 -bi486-linuxaout -s" ./configure
make cpio
make mt
```

ps and **free** The slackware source, for example, is *source/a/procps-1.01.tar.gz*. Unpack it, apply the patch found in the same directory, cd into the directory, and run

```
make CC="gcc -V2.7.2 -bi486-linuxaout -s" ps
make CC="gcc -V2.7.2 -bi486-linuxaout -s" free
```

e2fsck The slackware source, for example, is *source/a/e2fsbn/e2fsprogs-1.06.tar.gz*. Unpack it, cd into the directory, and run

```
mkdir build
cd build
CC="gcc -V2.7.2 -bi486-linuxaout -s" ../configure
make e2fsck
```

lilo The slackware source, for example, is *source/a/lilo/lilo.19.tar.gz*. Unpack it, cd into the directory, and run

```
make CC="gcc -V2.7.2 -bi486-linuxaout -s" free
```

login and **hostname** The slackware source, for example, is *source/a/util/util-linux-2.6.tar.gz*. Unpack it, cd into the directory, and edit the file *MCONFIG* to get the aout compiler:

```
CC=             gcc -V2.7.2 -bi486-linuxaout -s
```

then run the commands

```
cd login-utils
make login
cd ../misc-utils
make hostname
```

Finally, put all of them onto the floppy. Then edit the file */mnt2/etc/rc* so that you get the message you desire when you log in, also set the host name. The login message is in *issue*, so edit that, too. If you moved files from *etc sbin* you might also have to edit *inittab*. Also edit *fstab* to have the correct entries.

```
/dev/ram / ext2  defaults 1 1
/proc /proc proc defaults 0 0
/dev/hdc /cdrom iso9660 ro,suid,dev,exec,auto,async 1 1
/dev/hdd /cdrom iso9660 ro,suid,dev,exec,auto,async 1 1
```

Get the location of the darn *utmp* file correct, it should be in somewhere in */etc*, */var/adm*, or */var/run*, and linked from the other two places. The *wtmp* file needs to only appear in */etc* and */var/adm*.

After everything is set up to your liking, unmount the ram-disk, compress an image of it, and write it to the floppy at the appropriate place.

```
cd
umount /mnt2
dd if=/dev/ram2 bs=1024 count=4096 | gzip -9c >/root/tmp/rd.gz
dd if=/root/tmp/rd.gz of=dev/fd0 bs=1024 seek=500
```

Make sure that it fit onto the floppy! Now we have to tell the kernel on the floppy where to find the ram disk, and that it should load it at boot time.

```
rdev  /dev/fd0 /dev/fd0          # tell the kernel on /dev/fd0 that
                                 # the kernel is on /dev/fd0
rdev -r /dev/fd0 16884           # this tells the kernel to load the
                                 # ram-disk (set bit 14) located at block 500:
                                 # 16884=2^14+500
rdev -R /dev/fd0                 # tell the kernel to mount the root
                                 # system read/write enabled
```

This can also be done before you write the kernel to the floppy, the commands are then:

```
rdev zImage /dev/fd0
rdev -r zImage 16884
rdev -R zImage
```

Now reboot off the floppy. You should get some error message relating to the library versions the dynamic linker is using. Fix this as follows.

```
mount -t ext2 /dev/hda2 /mnt     # assuming your hard drive linux
                                 # partition is hda2
ldconfig -v
```

Now repeat the whole procedure of putting the image onto a ram disk, put that file *ld.so.cache* into it, and finish off as before.

```
zcat /mnt/root/tmp/rd.gz >/dev/ram2
mount -t ext2 /dev/ram2 /mnt2
cp /etc/ld.so.cache /mnt2/etc/ld.so.cache
umount /mnt2
dd if=/dev/ram2 bs=1024 count=4096 | gzip -9c >/mnt/root/tmp/rd.gz
dd if=/mnt/root/tmp/rd.gz of=dev/fd0 bs=1024 seek=500
```

Reboot and check that it works. In particular check *mt* and ftape if you want to use this disk to recover from disaster.

101 Right-Adjust

Files:
/usr/local/bin/right-adjust
/usr/local/src/right-adjust.c

`Right-adjust` is a utility that takes a text file and breaks the lines at existing white spaces so that each line is only 75 characters long. It makes a backup file of the source file by appending a tilde to the name. An optional second argument allows to specify a different number of characters per line than 75. This utility is usefull for ill-formatted e-mails.

101.1 right-adjust.c

```
/* Written by Uwe F. Mayer <mayer@tux.org> 1994 */

#include <stdio.h>
#include <string.h>
#include <ctype.h>

#define N 256

int main(argc, argv)
    int argc;
    char *argv[];
{
  FILE *in;
  FILE *out;
  int ch;
  int i;
  int line=0;
  int count=0;
  int limit=0;
  char buffer[N+1];
  int allowed=1;
  int space=0;
  int blankline=1;
  int lastch=0;


  if (argc>3 || argc==1)
    {
      printf("usage: right-adjust <filename> [number of columns]\n");
      return(1);
    }

  if (strlen(argv[1])>N-1)
    {
      printf("Filename is too long.\n");
```

```
    return(1);
  }

if (argc==3)
  {
    for(i=0;i<strlen(argv[2]);limit=limit*10+argv[2][i++]-'0');
  }
else
  {
    limit=72;
  }

if (limit>N)
  {
    printf("Chosen number of columns must be at most %d\n",N);
    return(1);
  }


for(i=0;i<strlen(argv[1]);i++)
  buffer[i]=argv[1][i];
buffer[i]='~';
buffer[i+1]=0;

if ((in=fopen(argv[1],"r"))==NULL)
  {
    printf("Cannot open %s to read\n",argv[1]);
    return(1);
  }

if ((out=fopen(buffer,"w"))==NULL)
  {
    printf("Cannot open backup file %s to write\n",buffer);
    return(1);
  }

while ((ch=getc(in)) != EOF)
  {
    putc(ch,out);
  }
fclose(in);
fclose(out);

if ((out=fopen(argv[1],"w"))==NULL)
  {
    printf("Cannot open %s to write\n",argv[1]);
    return(1);
  }

if ((in=fopen(buffer,"r"))==NULL)
  {
    printf("Cannot open backup file %s to read\n",buffer);
```

```
      return(1);
    }

  while ((ch=getc(in)) != EOF)
    {
      if (allowed)
{
  buffer[count++]=ch;
  blankline=blankline && isspace(ch); /* blankline=1 if blank line */
  if (count>limit)
    {
      if (line>0)
{
  line=0;
  putc('\n',out);
}
      for(i=0;i<count;putc(buffer[i++],out));
      allowed=0;
      space=isspace(ch); /* space=1 if there is a space */
    }
  else
    if (isspace(ch))
      {
if (!blankline && ch=='\n')
  {
    ch=' '; /* not a blank line, transform newline
       character into blank */
    buffer[count-1]=ch;
    blankline=1;
  }
if (blankline && ch=='\n')
  {
    if (lastch!='\n')
      {
putc('\n',out);
putc('\n',out);
      }
    else
      {
putc('\n',out);
      };
    lastch='\n';
  }
if (line+count>limit && count!=1)
  /* a single space does not open a new line */
  {
    line=0;
    putc('\n',out);
  }
if (count==1)
  count=0; /*avoid multiple spaces */
for(i=0;i<count;i++)
```

```
  {
    putc(buffer[i],out);
  }
if (count>0) lastch=ch;
line+=count;
count=0;
if (ch==’\n’)
  {
    line=0;
         }
      }
}
      else /* not allowed */
{
  if (ch==’\n’)
    {
      allowed=1;
      putc(ch,out);
      count=0;
      lastch=’\n’;
      blankline=1; /* a blank line so far */
    }
  else
    {
      space=isspace(ch)||space;
      if (!isspace(ch) && space)
{
  allowed=1;
  putc(’\n’,out);
  buffer[0]=ch;
  count=1;
  lastch=’\n’;
  blankline=0; /* not a blank line */
}
      else
{
  putc(ch,out);
}
    }
}
    }

  fclose(in);
  fclose(out);

  return(0);
}
```

102 Redhat Package Manager

Files:
*/usr/local/rpm/**

Sources obtainable at:
`http://www.redhat.com`
`ftp://sunsite.unc.edu/pub/Linux/distributions/redhat/code/rpm`

```
CFLAGS="-O2" ./configure --prefix=/usr/local/rpm
make
mkdir /usr/local/rpm
make install
```

Then compress the manpages, and link the binaries and the manual pages from the corresponding places in */usr/local*.

102.1 rpm2cpio

Files:
/usr/local/bin/rpm2cpio.pl

Sources obtainable at:
`http://www.iagora.com/ espel/rpm2cpio`

This is a replacement of the program *rpm2cpio* that comes with the rpm package. It is a Perl script, and can be installed alll by itself.

103 Save

Files:
/usr/local/bin/{save, play_save, mpeg_play, gv_save, xdvi_save}

This is a short script used to save standard input into a file. The file name is automatically generated using the time of the day. An optional argument to `save` can be used to specify an extension to this name. It will place the file into the subdirectory *tmp* of the users home directory. If this subdirectory doesn't exist it will be placed into the home directory itself. I use this script to automatically save any movies, postscript files, and sound files I visit with netscape, otherwise I would have to reload them if I wanted to see/hear them again.
save

```
#!/bin/sh
if [ -d ${HOME}/tmp ];
  then
    LOC=${HOME}/tmp
  else
    LOC=${HOME}
fi
```

```
if [ "$2" = "" ];
  then
    SUFFIX=.‘echo $1 | sed -e ’s/.*\.//’ -e ’s/\.//’‘
  else
    SUFFIX=.$2
fi
TIME=$(date ’+%D-%T’| tr ’/’ ’-’)
cat $1 > $LOC/tmp.$TIME$SUFFIX
chmod 600 $LOC/tmp.$TIME$SUFFIX
```

play_save

```
#!/bin/sh
save $1
cat $1 >/dev/audio
```

mpeg_save

```
#!/bin/sh
save $1
mpeg_play $1
```

gv_play

```
#!/bin/sh
save $1
ghostview $1
```

wavplay_save

```
#!/bin/sh
save $1
wavplay $1
```

xdvi_save

```
#!/bin/sh
save $1 dvi
xdvi $1
```

xanim_save

```
#!/bin/sh
save $1
xanim $1
```

104 Sc: Spreadsheet Calculator

Sources obtainable at:
http://sunsite.unc.edu/pub/Linux/distributions/slackware/source/ap

This is a standard slackware package. However, it has a year 2000 bug. Get the source, untar the file *sc-6.21.tar.gz*, and edit the files *interp.c* and *sc.doc* and change the number 1999 to 2037. The number 2037 is the year in which the numbers of seconds since 1970 will just not yet exceed the storage capacity of a 32 bit signed integer.

sc also allows to copy onto locked cells. By adding a check into *sc.c* at the 'c' command, it stops doing that. See the patch below.

Then pack the sources back up, and compile the package as usual, see Section 102.1.

The patch for sc version 6.21

```
diff -cr sc-6.21/interp.c sc-6.21-good/interp.c
*** sc-6.21/interp.c Mon May 25 18:35:50 1992
--- sc-6.21-good/interp.c Sat Jan 22 11:52:46 2000
***************
*** 484,490 ****
      mdays[1] = 28 + (yr%4 == 0);

      if (mo < 1 || mo > 12 || day < 1 || day > mdays[--mo] ||
!  yr > 1999 || yr < 1970) {
   error("@dts: invalid argument");
   cellerror = CELLERROR;
   return(0.0);
--- 484,490 ----
      mdays[1] = 28 + (yr%4 == 0);

      if (mo < 1 || mo > 12 || day < 1 || day > mdays[--mo] ||
!  yr > 2037 || yr < 1970) {
   error("@dts: invalid argument");
   cellerror = CELLERROR;
   return(0.0);
diff -cr sc-6.21/sc.c sc-6.21-good/sc.c
*** sc-6.21/sc.c Mon May 25 18:36:10 1992
--- sc-6.21-good/sc.c Sat Jan 22 12:25:16 2000
***************
*** 1245,1252 ****
       modflg++;
       for (c1 = curcol; arg-- && c1 < maxcols; c1++) {
   n = lookat (currow, c1);
!  (void) clearent(n);
!  copyent( n, p, currow - savedrow, c1 - savedcol);
       }
       break;
   }
--- 1245,1256 ----
       modflg++;
       for (c1 = curcol; arg-- && c1 < maxcols; c1++) {
   n = lookat (currow, c1);
!  if(n&&(n->flags&is_locked)){
!    error("Target cell locked. Not changed.");}
!  else {
!    (void) clearent(n);
```

```
!    copyent( n, p, currow - savedrow, c1 - savedcol);
!  }
       }
       break;
   }
diff -cr sc-6.21/sc.doc sc-6.21-good/sc.doc
*** sc-6.21/sc.doc Mon May 25 18:34:29 1992
--- sc-6.21-good/sc.doc Sat Jan 22 11:52:28 2000
***************
*** 2108,2114 ****
  .IP ""
  The month should be range from  1 to 12, the day should range from
  1 to the number of days in the specified month, and the year should
! range from 1970 to 1999.
  .\" ----------
  .TP 18
  .BR @tts (e1,e2,e3)
--- 2108,2114 ----
  .IP ""
  The month should be range from  1 to 12, the day should range from
  1 to the number of days in the specified month, and the year should
! range from 1970 to 2037.
  .\" ----------
  .TP 18
  .BR @tts (e1,e2,e3)
```

105 Secure Shell

Files:
*/usr/local/ssh1/**

Sources obtainable at:
`ftp://ftp.funet.fi/pub/unix/security/login/ssh/ssh-1.2.26.tar.gz`
`ftp://ftp.funet.fi/pub/crypt/cryptography/asymmetric/rsa/rsaref2.tar.gz`

Untar the first, and then `cd` into the newly created directory and untar the second. The file *login.c* has two bugs, the first line of

```
    ux.ut_syslen = strlen(ux.ut_host);
#ifdef HAVE_SYSLEN_IN_UTMPX
    ux.ut_syslen = strlen(ux.ut_host);
#endif
```

should be deleted. Also add the middle line of the segment below.

```
#ifdef HAVE_UTMPX_H
#define __USE_GNU
#include <utmpx.h>
```

Then compile and install with:

```
CFLAGS="-O2 -fomit-frame-pointer -D__USE_GNU" ./configure --with-rsaref \
--prefix=/usr/local/ssh1 --with-etcdir=/usr/local/ssh1/etc/
make
make install
```

Then remove the old installation, all files are kept with a *.old* extension. Compress the man pages, strip the binaries.

This procedure above only enables to use the client side. For a server one needs to have the secure shell daemon running. This needs messing with the booting files. Put a start command into the file */etc/rc.d/rc.M*.

106 Secure Shell Version 2

Files:
*/usr/local/ssh2/**
*/etc/ssh2/**
*~/.ssh2/**

Sources obtainable at:
`http://www.ssh.fi`

The sources need a bit of editing. In the file *pty-ptmx.c* comment out the following:

```
#include <sys/stream.h>
#include <sys/conf.h>
```

The file *login.c* has two bugs, add the following switch:

```
#ifdef HAVE_SYSLEN_IN_UTMPX
    ux.ut_syslen = strlen(ux.ut_host);
#endif
```

Also add the middle line of the segment below.

```
#ifdef HAVE_UTMPX_H
#define __USE_GNU
#include <utmpx.h>
```

Compile with:

```
CFLAGS="-O3 -fomit-frame-pointer -D__USE_GNU" ./configure --with-rsaref \
--prefix=/usr/local/ssh2 --with-etcdir=/usr/local/ssh2/etc/
make
make install
```

Then remove the old installation, all files are kept with a *.old* extension. Compress the man pages, strip the binaries. The link everything from */usr/local*. I left *ssh1* as the default *ssh*.

This procedure above only enables to use the client side. For a server one needs to have the secure shell daemon running. This needs messing with the booting files. Put a start command into the file */etc/rc.d/rc.M*.

107 Seyon

Files:
/root/.seyon/{*phonelist, protocols, script.CIS, script.PCBoard, script.QWK, script.UTAH, script.unix, startup*}
/root/.Xmodmap

`Seyon` installs right off the CD ROM. Just a few initialization files need to be modified. Those are kept in the *./seyon* subdirectory with the exception of *.Xmodmap* which specifies how the window of `seyon` comes up in X. Here is the relevant part.

```
Seyon.modems:                   /dev/ttyS3
Seyon.showFunMessages:          off
Seyon.vt100.background:         black
Seyon.vt100.foreground:         green
Seyon.vt100.geometry:           80x25+125-10
Seyon.vt100.saveLines:          100
```

`.seyon/startup` contains the initialization information for the modem:

```
set baud 38400
transmit "AT&C1&Q5W1S0=0S36=7S46=138S48=7^M"
# initialization string
```

`.seyon/phonelist` contains (you guessed it) a list of phone numbers and corresponding scripts. Here is an example script.

```
# The following IF condition tests to see if we are linked from a .phonelist
# entry.  If not, we manually dial the annex building:
if !linked
  dial AnnexPhoneNumber   # Note anything past a full command is ignored
endif
# Wait for a CONNECT message from the modem.  If we time out (60 seconds),
# redial the number (it is probably busy).
connect:
waitfor "CONNECT" 60
if !waitfor
redial
goto connect
   endif
# Wait to let the annex catch up with us
waitfor "prompt"
waitfor "Username: "
# Transmit the user id string (remember the ^M at the end for ENTER)...
```

```
transmit "YourAnnexUserName^M"
# Wait for cis server to ask for a password...
waitfor "Password: "
# Transmit our password
transmit "YourAnnexPassWord^M"
# Wait for cis server command prompt
waitfor "> "
transmit "telnet workstation.internet.wherever^M"
# Wait for workstation
waitfor "login: "
transmit "WorkStationUserName^M"
waitfor "Password: "
transmit "WorkStationPassWord^M"
waitfor "(vt100) "
transmit "^M"
```

108 SGML

Files:
/usr/local/bin/{fixref, genertoc, html2html, rtf2rtf, sgml2html, sgml2info, sgml2latex, sgml2lyx, sgml2rtf, sgml2txt, sgmlcheck, sgmls, sgmlsasp}
/usr/local/man/man1/{sgml2html.1.gz, sgml2info.1.gz, sgml2latex.1.gz, sgml2lyx.1.gz, sgml2rtf.1.gz, sgml2txt.1.gz, sgmls.1.gz, sgmlsasp.1.gz}
*/usr/local/doc/sgml-tools/**
*/usr/local/lib/sgml-tools/**
*/usr/local/lib/texmf/tex/latex/sgml/**

Sources obtainable at:
`http://www.xs4all.nl/~cg/sgmltools/`
`http://sunsite.unc.edu/pub/Linux/utils/text/sgml-tools-0.99.0.tar.gz`

Unpack the sources, cd into the created directory and run `./install`. This will ask a few questions, place all the stuff as it is suggested, however, perl is at */usr/bin/perl* and not at */usr/local/bin/perl* on linux systems. Then compress the man pages, and link all the LaTeX files */usr/local/lib/sgml-tools/*.{sty,tex}* from */usr/local/texmf/tex/latex/sgml/* so that LaTeX can actually find them. Finally remove a few versions of the guide in */usr/local/doc/sgml-tools*, I decided to keep the *sgml*, the *dvi*, and the *html* versions.

109 GNU Shell Utilities

Files:
/bin/{ date echo hostname pwd su uname false true }
/usr/sbin/chroot
/usr/bin/{ basename dirname env expr factor id logname pathchk printenv printf seq sleep tee test tty users who whoami yes nice uptime groups nohup }
/usr/info/sh-utils.info.gz
/usr/man/man1/{ hostname.1.gz true.1.gz id.1.gz uname.1.gz tty.1.gz logname.1.gz users.1.gz

pathchk.1.gz who.1.gz printenv.1.gz whoami.1.gz printf.1.gz yes.1.gz pwd.1.gz nice.1.gz sleep.1.gz nohup.1.gz tee.1.gz su.1.gz test.1.gz stty.1.gz basename.1.gz chroot.1.gz date.1.gz dirname.1.gz echo.1.gz env.1.gz expr.1.gz false.1.gz groups.1.gz }

Sources obtainable at:
`http://sunsite.unc.edu/pub/gnu`

```
CFLAGS="-O2" ./configure --prefix=/usr --disable-nls
make
make install
```

Notice that I disabled the native language support. Compress all the man and info pages. The install script installs all binaries into */usr/bin*, but Linux has some of them in */bin* and one in */usr/sbin*. You need to move those manually. Also, strip all the binaries.

110 SlackBuild and Pkgtool

SlackBuild is a shell script that is found in all Slackware source directories, and it can be executed by the shell to build the current package. It dumps the final package into the directory pointed to by the variable TMP, and if that is not set, into */tmp*. I prefer to have things happening where I want them. For example, to build the *xfractint* package:

```
export TMP=where_ever_I_want_it
cd /usr/src/slakware/source/xap/xfractint/
umask 022
sh SlackBuild
```

After it's done, the package, once expanded and once as a gzipped tar archive, and the build tree all are found in the directory *build*. Of course, if not run as root, it will have the wrong ownership in the tar archive. Hence it is better to complete with:

```
cd where_ever_I_want_it
cd package-xfract
chown -R root.root *
tar -zcvf ../xfract.tgz *
```

You noticed that the package is not called *xfractint* but *xfract*, to keep the name shorter that nine digits. Then get from the corresponding binary distribution directory the necessary installation description file, that is here (because xfracint is in the xap1 directory) the file *diskxap1*. Finally run *pkgtool*, it will complain that there are packages missing, but it will allow you to install the newly comiled package. Of course, you could have simply untarred from the root directory, but then you won't get the file *var/log/packages/xfract*, which makes it so much easier to uninstall. The files *diskap1* contains the entry below. Such an entry could be written for any *.tgz* file which untars from */*, just make sure you also have the leading directories in the tar archive. The CONTENTS line may be as long as it needs to be. The entries describing the package must start at the eleventh column.

```
CONTENTS: fvwm95 xfract xgames xpaint xpdf xspread xv xvim xxgdb
xfract:   xfractint-3.04
```

```
xfract:
xfract:   Xfractint draws a wide variety of fractals. Xfractint is a port of
xfract:   the DOS program Fractint.  The primary original authors of Fractint
xfract:   are Bert Tyler, Timothy Wegner, Mark Peterson, and Pieter
xfract:   Branderhorst.  The Xfractint port is by Ken Shirriff.
xfract:
```

Finally, it might be necessary to run some commands to finish off the installation. These should be collected in a shell script (without the usual `#!/bin/sh` header) called *install/install.sh*. This script should be part of the *.tgz* archive, and will be executed and stored by *pkgtool* under the name of the package in the */var/adm/scripts* directory.

111 Slackware Setup

Preparation

Running DOS make yourself a directory, say, *C:/linux*. Copy the files */MCC/dos-utils/gzip124.exe* and */MCC/dos-utils/rawrite.exe* from the CD-ROM into this directory. Then copy the appropriate boot-disk and root-disk image into this directory, for me it is */slackwar/bootdsks.144/idecd.gz* and */slackwar/rootdsks.144/color144.gz*. Uncompress them with `gzip124 -d color144` and `gzip124 -d idecd`. The run `rawrite`, it asks for the file name (*idecd* or *color144*) and the drive letter (*a:*). Then put the boot-disk in and reboot. Hit `Enter` when necessary and enter the root-disk when asked. Now run `fdisk`, see Section . Do not prepare your swap partitions, you can do it from within `setup`. Then finally run `setup`.

Using setup

`Setup` is nicely menu-driven. Start with with the swap partitions. Go with the flow: I picked the source of the installation as the Slackware CD, with a normal hardware installation (slakware). I made the DOS partition visible under Linux.

I selected to install from the following disk sets: A, AP, D, DES, E, GTK, K, KDE, N, T, TCL, X, XAP, and XV. I decided to skip the following disk set: F, XD, and Y. This is the default choice with the exception of skipping F and Y. I can always read the extra documentation from the CD later (that's series F), and I don't play text-based games (series Y), and I do not develop for X (series XD).

I chose to install in expert mode. I made a few changes to the suggested packages. In the lists below, [ ] stands for not installed against the default, while [X] stands for additionally installed.
A: [ ] scsi, [ ] isapnp, [ ] loadlin, [ ] minicom, [ ] umsprogs.
AP: [X] bc, [X] enscript, [X] ispell, [ ] jed, [ ] joe, [ ] quota, [ ] raidtool, [X] rpm, [X] sc, [ ] vim, [ ] zsh.
D: [ ] autoconf, [ ] automake, [X] egcs_g77, [ ] pmake, [ ] slang.
DES:
E: [X] emac_nox.
GTK: [ ] audiofil, [ ] control, [ ] econf, [ ] ee, [ ] enlight, [ ] esound, [ ] eterm, [ ] fnlib, [ ] gdkpixbf, [ ] gdm, [ ] gedit, [ ] gftp, [ ] ggv, [ ] gmc, [ ] gmp, [ ] gnoadmin, [ ] gnoaudio, [ ] gnogames, [ ] gnoguide, [ ] gnomaps, [ ] gnomcore, [ ] gnomedia, [ ] gnomeicu, [ ] gnomenet, [ ] gnomepim,

[ ] gnometop, [ ] gnomlibs, [ ] gnomobjc, [ ] gnoprint, [ ] gnotepad, [ ] gnoutils, [ ] gnpython, [ ] gnumeric, [ ] gqmpeg, [ ] gtkeng, [ ] guile, [ ] libghttp, [ ] libglade, [ ] libgtop, [ ] librep, [ ] libxml, [ ] orbit, [ ] repgtk, [ ] sawfish, [ ] wmaker, [ ] xchat, [ ] xscrsave.
K:
KDE: [ ] kdegames, [ ] kgraphic, [ ] kmedia, [ ] knetwork, [ ] korganiz.
N: [ ] apache, [X] dip, [ ] inn, [ ] tcpdump, [ ] tin, [ ] trn.
T: [X] transfig, [X] xfig.
TCL: [X] expect.
X: [ ] xxfb, [ ] lesstif.
XAP: [ ] fvwm2, [ ] fvwmicns, [ ] fvwm95, [X] gnuplot, [ ] imagick, [ ] xgames, [ ] xfract, [ ] xvim, [ ] xxgdb.
XV:

Then I installed the kernel from the bootdisk as the default kernel. I skipped making a bootdisk. I linked the modem to */dev/ttyS0*. I do not want custom screen fonts.
I chose the expert mode for LILO configuration: Start with Begin, then choose the Standard screen, add a linux bootable partition "linux" for */dev/hda2*, then another one "backup" for */dev/hda2* as well (edit that later to point to the backup kernel), and finally add the DOS partition */dev/hda1* with label "win98". Then write the file.
Network configuration (can be fixed later with "netconfig"):
hostname: tosca
domain: frop.org
connection: loopback
Mouse configuration: ps2. I do not want "gpm -t ps2 -m /dev/mouse" to run at startup.
Sendmail configuration: SMTP+BIND
Default X-Window manager: xinitrc.kde

112 Sound

Files:
/usr/src/linux/Create_sound_drivers
/usr/src/linux/Configure_sound_card
/usr/local/bin/play

This is mostly obsolete information.

Getting the sound card to work involves essentially two steps. First create the necessary special device files, and then compile the kernel with sound enabled. If you are upgrading from an older sound driver, make sure that */usr/include/sys/soundcard.h* and */usr/include/sys/ultrasound.h* are symbolic links to the corresponding files in */usr/include/linux*. `Create_sound_drivers` is the script that does the first task. You have to run it before you compile the kernel. I copied it from the sound HOWTO.

.au files can simply be played with `cat` file `> /dev/audio`. In fact I made this into the script *play*.

```
#!/bin/sh
cat $1 >/dev/audio
```

The kernel configuration, stored in */etc/soundconf*

```
/* Generated by configure. Don't edit!!!! */
/* Making changes to this file is not as simple as it may look. */

/* If you change the CONFIG_ settings in local.h you */
/* _have_ to edit .defines too. */

#undef  CONFIG_PAS
#undef  CONFIG_SB
#define CONFIG_ADLIB
#undef  CONFIG_GUS
#undef  CONFIG_MPU401
#undef  CONFIG_UART6850
#undef  CONFIG_PSS
#undef  CONFIG_GUS16
#undef  CONFIG_GUSMAX
#define CONFIG_MSS
#undef  CONFIG_SSCAPE
#undef  CONFIG_TRIX
#undef  CONFIG_MAD16
#undef  CONFIG_CS4232
#undef  CONFIG_MAUI
#undef  CONFIG_SPNP
#undef  CONFIG_UNUSED1
#undef  CONFIG_UNUSED2
#undef  CONFIG_AEDSP16
#define CONFIG_AUDIO
#undef  CONFIG_MIDI
#define CONFIG_YM3812

#undef  CONFIG_MPU_EMU
#define CONFIG_AD1848
#undef  CONFIG_SBDSP
#undef  CONFIG_UART401
#define CONFIG_SEQUENCER

#define MSS_BASE 0x530
#define MSS_IRQ 10
#define MSS_DMA 1
#define DSP_BUFFSIZE 65536
#define SELECTED_SOUND_OPTIONS 0x00880204
```

113 StarOffice

Files:
/glibc2/soffice/{libICE.so, libICE.so.6, libSM.so, libSM.so.6, libX11.so, libX11.so.6, libXext.so, libXext.so.6, libXmu.so, libXmu.so.6, libXt.so, libXt.so.6}
*/usr/local/Office51/**

~/.sversionrc
~/.sorc
*~/Office51/**

Sources obtainable at:
`ftp://www.stardivision.com`
`ftp://sunsite.utk.edu/pub/StarDivision/unxlnxi/so50_01.tar`
`ftp://narnia.mit.edu/pub/StarOffice/unxlnxi/so50_01.tar`
`http://sunsite.unc.edu/StarDivision/unxlnxi/so50_01.tar`
`http://sunsite.utk.edu/StarDivision/unxlnxi/so50_01.tar`

Unpack the sources. Staroffice needs the glibc2 C-library and other libraries, they come with staroffice in the directory *glibc2_inst* as part of the distribution.

```
cd /
tar -zxvf glibc2.tar.gz
cd lib
ln -s ../glibc2/ld-linux.so.2
ldconfig -v
```

If you already have *glibc2* you don't need to do any of the above. In particular, I replaced the version 2.0.7 glibc files wiith a newer version, and kept only those that I had no newer version of. Now, run as root `./setup /net` (note the space before `/net`) from the *so51inst* subdirectory. This is very much Microsoft like, just follow the on-screen instructions. Put */usr/local/Office51/bin* into the path in */etc/profile*. Every single user then must run */usr/local/Office51/bin/setup*, which copies another 2 MB worth of stuff into each user's home directory. If desired, make an entry in */usr/X11/lib/X11/fvwm/system.fvwmrc* or for whatever window manager you use. You should also add the fonts from StarOffice to the X server's font path by editing */usr/X11/lib/X11/XF86Config*. Add the lines below.

```
FontPath   "/usr/local/Office51/fonts/type1/"
FontPath   "/usr/local/Office51/fonts/75dpi/"
```

Java suppport

StarOffice needs a java runtime environment that supports native threads, see Section 51.1.5. This can be chosen during installation of StarOffice, or later by running the command *jvmsetup*. It can also be set up later, by clicking Explorer – Bookmarks – Java – Java Setup. You need to specify the directory in which the runtime environment is installed.

Registration information

This was only needed up to version 5.0, starting with version 5.1 you no longer need a registration key.

114 SVGA Library

Files:
/usr/lib/{libvga.a, libvga.sa, libvga.so.1–>libvga.so.2.10, libvga.so.1.2.10, libvga.so.1.2.7,

libvgagl.a, libvgagl.sa, libvgagl.so.1–>libvgagl.so.1.2.10, libvgagl.so.1.2.10}
/usr/include/{*vga.h, vgakeyboard.h, vgamouse.h*}

The sources are available via ftp at `sunsite.unc.edu/pub/Linux/lib/graphics/`. Unpack the sources and edit the files *Makefile.cfg* and *libvga.config*. Then run a `make shared` and install the files listed above by hand, and then run *ldconfig* to update the dynamic linker datafile. Any binary using this library must be SUID root, or you'll get an error saying that the program cannot get I/O permissions.

Makefile.cfg

```
#----------------------------------------------------------------------
# SVGAlib Compile-time configuration file
#----------------------------------------------------------------------

MAJOR_VER = 1
MINOR_VER = 2.10
VERSION = $(MAJOR_VER).$(MINOR_VER)

#----------------------------------------------------------------------
# Configuration Section
#----------------------------------------------------------------------

# Source directory.
#srcdir = /usr/local/src/svgalib-$(VERSION)
srcdir = $(shell sh -c pwd)

# Common prefix for installation directories.
# NOTE: This directory must exist when you start the install.
prefix = /usr/local
exec_prefix = $(prefix)

# Directory where the shared stubs and static library will be installed.
libdir = $(exec_prefix)/lib

# Directory where the shared library will be installed.
sharedlibdir = /lib

# Directory where the font and textmode utilities will be installed.
bindir = $(exec_prefix)/bin

# Directory where the run-time configuration files will be installed.
datadir = /etc/vga

# Directory where the header files will be installed.
includedir = $(prefix)/include

# Target binary format.
#TARGET_FORMAT = a.out
TARGET_FORMAT = elf
```

```
# Uncomment this if you want to compile and install the static libs.
#INSTALLSTATICLIB = installstaticlib

# Comment this out if you don't want to install the shared libs.
INSTALLSHAREDLIB = installsharedlib

# Comment this out if you don't want to compile and install the utilities.
#INSTALLUTILS = installutils

#
# Comment out any driver that you don't want included in the library.
#
#INCLUDE_ET4000_DRIVER = y
#INCLUDE_CIRRUS_DRIVER = y
#INCLUDE_TVGA_DRIVER = y
#INCLUDE_OAK_DRIVER = y
INCLUDE_EGA_DRIVER = y
#INCLUDE_MACH32_DRIVER = y
INCLUDE_S3_DRIVER = y
#INCLUDE_ET3000_DRIVER = y
#INCLUDE_GVGA6400_DRIVER = y
#INCLUDE_ARK_DRIVER = y
#INCLUDE_ATI_DRIVER = y
#INCLUDE_ALI_DRIVER = y
#INCLUDE_MACH64_DRIVER = y

#
# Comment out any adapter you don't want to autodetect.
#
#INCLUDE_ET4000_DRIVER_TEST = y
#INCLUDE_CIRRUS_DRIVER_TEST = y
#INCLUDE_TVGA_DRIVER_TEST = y
#INCLUDE_OAK_DRIVER_TEST = y
INCLUDE_EGA_DRIVER_TEST = y
#INCLUDE_MACH32_DRIVER_TEST = y
#INCLUDE_GVGA6400_DRIVER_TEST = y
INCLUDE_S3_DRIVER_TEST = y
#INCLUDE_ET3000_DRIVER_TEST = y
#INCLUDE_ARK_DRIVER_TEST = y
#INCLUDE_ATI_DRIVER_TEST = y
#INCLUDE_ALI_DRIVER_TEST = y
#INCLUDE_MACH64_DRIVER_TEST = y

#
# Comment out any dac support that you don't want included in the library.
#
INCLUDE_NORMAL_DAC = y
INCLUDE_S3_SDAC_DAC = y
INCLUDE_S3_GENDAC_DAC = y
INCLUDE_S3_TRIO64_DAC = y
#INCLUDE_SIERRA_DAC = y
#INCLUDE_SC15025_DAC = y
```

```
#INCLUDE_ATT20C490_DAC = y
#INCLUDE_ATT20C498_DAC = y
#INCLUDE_ICW_DAC = y
#INCLUDE_IBMRGB52x_DAC = y
#INCLUDE_SC1148X_DAC = y

#
# Comment out any dac you don't want to autodetect.
# (not all dacs can be autodetected, at this time)
#
INCLUDE_S3_SDAC_DAC_TEST = y
INCLUDE_S3_GENDAC_DAC_TEST = y
#INCLUDE_SC15025_DAC_TEST = y
#INCLUDE_ATT20C490_DAC_TEST = y
#INCLUDE_IBMRGB52x_DAC_TEST = y
#INCLUDE_SC1148X_DAC_TEST = y

# Location of the svgalib configuration file.
SVGALIB_CONFIG_FILE = $(datadir)/libvga.config

# Defining DYNAMIC enables runtime parsing of the file defined by
# ET4000_REGS (usually /etc/libvga.et4000) for the et4000
# driver. See et4000/README for details. Commenting this out again
# saves binary space.
#
# If you just want to use the et4000.regs in the source directory,
# comment out the definition of DYNAMIC.
#DYNAMIC = y
#ET4000_REGS = $(datadir)/libvga.et4000

# The EGA driver may load additional modes (SuperEGA cards) like the
# et4000 driver does. Just define the configuration file below.
# [This should be taken with a grain of salt, EGA is untested.]
#EGA_REGS = $(datadir)/libvga.ega

# Defining USE_CLOCKS will cause the ET4000 driver to measure clock
# frequencies (they are not actually used yet).
#USE_CLOCKS = y

# Uncomment to allow mouse type overrides
ALLOW_MOUSE_OVERRIDE = y

#-----------------------------------------------------------------------
# Compiler Section
#-----------------------------------------------------------------------

# Compiler used.
CC       = gcc
#CC      = gcc -b i486-linuxaout
#CC      = gcc -b i486-linux

# You might want to add -m386 here if you have a recently installed
```

```
# (486 configured) compiler on a 386. The only real difference is the
# generous alignment padding of function entry-points for the 486.
WARN     = -Wall -Wstrict-prototypes
INCLUDES = -I$(srcdir)/include -I.
OPTIMIZE = -fomit-frame-pointer -O2 -fno-strength-reduce -pipe
CFLAGS   = $(WARN) $(DLLFLAGS) $(INCLUDES) $(OPTIMIZE) $(DEFINES)
LDFLAGS   = -N -s

# Uncomment the following if you are compiling a.out shared libraries
# with an ELF ld.
#
#MKIMAGEFLAGS = -m i386linux -oformat a.out-i386-linux -qmagic

# additional flags for shared lib.
ifeq (a.out, $(TARGET_FORMAT))
DLLFLAGS = -B/usr/dll/jump/
else
DLLFLAGS = -fPIC
endif

# Utilites used.
AR   = ar
INSTALL_PROGRAM = install -c -s -m 755 -o root -g bin
INSTALL_SHLIB = install -c -m 755 -o root -g bin
INSTALL_DATA = install -c -m 644 -o root -g bin
```

libvga.config

```
# Configuration file for svgalib. Default location is /etc/vga.
# Lines starting with '#' are ignored.

# Have a deep look at README.config to see what can do here (especially
# for mach32).

# Mouse type:

# mouse Microsoft # Microsoft
# mouse MouseSystems # Mouse Systems
# mouse MMSeries # Logitech MM Series
# mouse Logitech # Logitech protocol (old, newer mice use Microsoft protocol)
# mouse Busmouse # Bus mouse
# mouse PS2 # PS/2 mouse
# mouse MouseMan # Logitech MouseMan
# mouse none # None

mouse MouseSystems

# Some multiprotocol mice will need one of the following:

# setRTS   # set the RTS wire.
# clearRTS # clear the RTS wire.
# leaveRTS # leave the RTS wire alone (default) (Wire is usually set)
```

```
# setDTR   # set the DTR wire.
# clearDTR # clear the DTR wire.
# leaveDTR # leave the DTR wire alone (default) (Wire is usually set)

# Monitor type:

# Only one range can be specified for the moment.  Format:
# HorizSync min_kHz max_kHz
# VertRefresh min_Hz max_Hz

# Typical Horizontal sync ranges
# (Consult your monitor manual for Vertical sync ranges)
#
# 31.5 - 31.5 kHz (Standard VGA monitor, 640x480 @ 60 Hz)
# 31.5 - 35.1 kHz (Old SVGA monitor, 800x600 @ 56 Hz)
# 31.5 - 35.5 kHz (Low-end SVGA, 8514, 1024x768 @ 43 Hz interlaced)
# 31.5 - 37.9 kHz (SVGA monitor, 800x600 @ 60 Hz, 640x480 @ 72 Hz)
# 31.5 - 48.3 kHz (SVGA non-interlaced, 800x600 @ 72 Hz, 1024x768 @ 60 Hz)
# 31.5 - 56.0 kHz (high frequency, 1024x768 @ 70 Hz)
# 31.5 - ???? kHz (1024x768 @ 72 Hz)
# 31.5 - 64.3 kHz (1280x1024 @ 60 Hz)

HorizSync 30 64
VertRefresh 50 100

# Montitor timings
#
# These are prefered over the default timings (if monitor and chipset
# can handle them). Not all drivers use them at the moment, and Mach32
# has its own syntax (see below).
# The format is identical to the one used by XFree86, but the label
# following the modeline keyword is ignored by vgalib.
#
# Here some examples:

# modeline "640x480@100"  43  640  664  824  848   480  483  490  504
# modeline "800x600@73"   50  800  856  976 1024   600  637  643  666
# modeline "1024x768@75"  85 1024 1048 1376 1400   768  771  780  806

modeline  "640x480@72"   31      640  664  704  832    480  489  492  520
modeline "800x600" 50.00  800 856 976 1040  600 617 623 642 +hsync +vsync
modeline "1024x768"      75 1024 1048 1184 1328 768 771 777 806 -hsync -vsync
modeline "1280x1024"     110    1280 1360 1552 1724   1024 1031 1034 1060
modeline "800x600@24bit"  50     800  836  956 1024    600  637  643  666


# Mach32 timings:

# e.g. Setup a 320x200 mode for the mach32:

#define 320x200x32K 320x200x64K 320x200x16M 320x200x16M32
#                     16 320 392 464 552 200 245 265 310
```

```
# These are REQUIRED for above mode, please edit to suit your monitor.
# (No, I won't pay for a new one)
# HorizSync 29 65
# VertRefresh 42 93.5

# Chipset type:
#
# Use one of the following force chipset type.
# Autodetects if no chipset is specified.
# If a chipset driver gives trouble, try forcing VGA.

# chipset VGA # Standard VGA
# chipset EGA # EGA
# chipset ET3000 # Tseng ET3000
# chipset ET4000 # Tseng ET4000
# chipset Cirrus # Cirrus Logic GD542x
# chipset TVGA # Trident TVGA8900/9000
# chipset Oak # Oak Technologies 037/067/077
chipset S3 # S3 chipsets
# chipset GVGA6400 # Genoa 6400
# chipset ARK # ARK Logic
# chipset ATI # old ATI VGA
# chipset Mach32 # ATI Mach32
# chipset ALI # ALI2301
# chipset Mach64 # ATI Mach64

# EGA Color/mono mode:
# Required if chipset is EGA.
#
# Use one of the following digits to force color/mono:

# monotext  # Card is in monochrome emulation mode
# colortext # Card is in color emulation mode
colortext

# RAMDAC support:
# Some chipsets (e.g. S3 and ARK) allows specifying a RAMDAC type.
# If your RAMDAC is not autodetected, you can try specifying it.
# Do NOT specify a RAMDAC if you card uses the S3 Trio chipset
# (the RAMDAC is built in).

# Ramdac Sierra32K
# Ramdac SC15025
# Ramdac SDAC         # S3 SDAC
# Ramdac GenDAC       # S3 GenDAC
# Ramdac ATT20C490    # AT&T 20C490, 491, 492 (and compatibles)
# Ramdac ATT20C498    # AT&T 20C498
# Ramdac IBMRGB52x    # IBM RGB524, 526, 528 (and compatibles)

# Dotclocks:
# Some chipsets needs a list of dot clocks for optimum operation.  Some
```

```
# includes or supports a programmable clock chip.  You'll need to specify
# them here.

# Fixed clocks example:
# (The following is just an example, get the values for your card from
#  you XF86Config)

# Clocks 25.175 28.3 40 70 50 75 36 44.9 0 118 77 31.5 110 65 72 93.5

# Programmable clockchip example:

# Clockchip ICD2061A  # The only one supported right now
```

115 Tar

Files:
/bin/static/{tar, rmt}
/usr/bin/tar
*/usr/info/tar.info**
/usr/libexec/rmt
/usr/man/man1/tar.1.gz
/usr/man/man8/rmt.8.gz

Sources obtainable at:
`ftp://ftp/gnu.ai.mit.edu/pub/gnu/tar/`

```
./configure --prefix=/usr --disable-nls
make CFLAGS="-O2 -s"
make check
make install
gzip -9 /usr/info/tar*
rm src/tar src/rmt
make CFLAGS="-O2 -s -static"
ginstall -m 755 -o root -g bin tar rmt /bin/static
```

Don't delete the old man pages, GNU tar comes now only with info documentation. I specify the `--disable-nls` flag because I don't want the international language support. The binaries in */bin/static* are for disaster recovery.

116 Tcl/Tk

Files:
*/usr/local/tcl/**
*/usr/local/tk/**

Sources obtainable at:
`http://www.scriptics.com/products/8.0.html`

Preparation

I want compressed man pages, and this is somewhat of a hassle here as many pages have more than one hard link to them and can therefore not be compressed. The thing to do is to edit *mkLinks* so that it expects compressed man pages names (just replace .3 by .3.gz and .n by .n.gz using `emacs` or `sed` in this script), and of course *mkLinks* needs to get a line added, so that it first compresses the man pages. This needs to be done both for tcl and for tk. This is not all of it, because the man pages are not installed compressed, but uncompressed instead. Tcl needs to compiled first. Finally, also install the private header files, *dejagnu* wants them.

Tcl

```
cd unix
./configure --prefix=/usr/local/tcl
make CFLAGS="-O3"
make install
# now install the private header files
cd ..
for i in ‘find unix generic -name tcl\*.h‘ ;do
  install -m 644 $i /usr/local/tcl/include
done
# other odds and ends
cd /usr/local/tcl/bin
strip tclsh8.0
ln -s tclsh8.0 tclsh
gzip -9 ../man/man1/*.1
# now link the files from /usr/local
ln -s /usr/local/tcl/bin/* /usr/local/bin/
ln -s /usr/local/tcl/lib/* /usr/local/lib/
ln -s /usr/local/tcl/man/man1/* /usr/local/man/man1/
ln -s /usr/local/tcl/man/man3/* /usr/local/man/man3/
ln -s /usr/local/tcl/man/mann/* /usr/local/man/mann/
```

Tk

```
cd unix
./configure --prefix=/usr/local/tk
make CFLAGS="-O3"
make install
# now install the private header files
cd ..
for i in ‘find unix generic -name tk\*.h‘ ;do
  install -m 644 $i /usr/local/tk/include
done
# other odds and ends
cd /usr/local/tk/bin
strip *
ln -s wish8.0 wish
gzip -9 ../man/man1/*.1
# now link all the files from /usr/local
ln -s /usr/local/tk/bin/* /usr/local/bin/
```

```
ln -s /usr/local/tk/lib/* /usr/local/lib/
ln -s /usr/local/tk/man/man1/* /usr/local/man/man1/
ln -s /usr/local/tk/man/man3/* /usr/local/man/man3/
ln -s /usr/local/tk/man/mann/* /usr/local/man/mann/
# finally update the whatis database of manual pages
makewhatis -v /usr/local/man
```

117 The teTEX Distribution

Files:
*/usr/local/teTeX/**

Sources obtainable at:
`http://sunsite.unc.edu/pub/Linux/apps/tex/teTeX/1.0/distrib/`
`binaries/ix86-linux-libc5.tar.bz2`
`doc/man.tar.bz2`
`doc/info.tar.bz2`
`fonts/misc-type1.tar.bz2`
`fonts/yandy.tar.bz2`
`fonts/blackboard.tar.bz2`
`fonts/misc-mf.tar.bz2`
`fonts/sauter.tar.bz2`
`fonts/ae.tar.bz2`
`fonts/ec.tar.bz2`
`fonts/ams.tar.bz2`
`fonts/concrete.tar.bz2`
`base/tetex-base.tar.bz2`
`base/latex-base.tar.bz2`
`goodies/latex-extra.tar.bz2`
`goodies/drawing.tar.bz2`
`goodies/misc-macros.tar.bz2`
`goodies/fontinst.tar.bz2`

Install as follows:

```
umask 022
mkdirhier /usr/local/teTex/bin/ix86-pc-linux-gnulibc1
mkdirhier /usr/local/teTex/share/texmf
mkdirhier /usr/local/teTex/man
mkdirhier /usr/local/teTex/info
umask 0
cd /usr/local/teTeX/bin/ix86-pc-linux-gnulibc1
bunzip2 < binaries/ix86-linux-libc5.tar.bz2 | tar -xvf -
cd ../../man
bunzip2 < doc/man.tar.bz2 | tar -xvf -
cd ../info
bunzip2 < doc/info.tar.bz2 | tar -xvf -
```

```
cd ../share/texmf
# ...
# now unpack all the others from the list of sources above
# ...
cd doc
find . name \*.ps -o -name \*.dvi | xargs gzip -9
./mkhtml > helpindex.html
```

Edit */etc/profile* to add the teTEX distribution to the PATH and MANPATH, also add the documentation to */usr/info/dir* and to */usr/local/html/index.html*. Compress the man and info pages. Edit the binary *mktexpk* to add the modes you want:

```
    196) MODE=gtfaxl;;
    204) MODE=gtfax;;
    360) MODE=bjtenex;;
```

Edit *share/texmf/dvips/config/config.ps* and comment out the lpr statement, so that by default the Postscript file goes to a file and not to the printer. Now comes the post-installation from the teTeX instructions itself.

```
#     make sure that the main texmf.cnf files is correctly found:
kpsewhich texmf.cnf
#     If that file is not found correctly, you have to set the TEXMFCNF
#     environment variable (again with export or setenv), e.g. like this:
#   TEXMFCNF=/usr/local/teTeX/share/texmf/web2c; export TEXMFCNF
#     make sure that TEXMFMAIN expands correctly:
kpsewhich --expand-var='$TEXMFMAIN'
#     That should give you the name of the main texmf tree, e.g.
#   /usr/local/teTeX/share/texmf
#     If it does not, please change the TEXMFMAIN definition in
#     the texmf.cnf (see previous step).
#     run texhash and texconfig and change what you need:
texhash
texconfig
#     You are done!
```

The information below is no longer current.

`http://sunsite.unc.edu/pub/Linux/apps/tex/teTeX/distrib/sources/` is the location of the sources. This is the TEX distribution nowadays used with Slackware. I got the sources and compiled them myself. Linux places TEX into */usr/lib*, while the teTEX distribution is by default placed into */usr/local/*. First, get the `lib` archive and unpack from its final destination. Then get the source file and place just about anywhere for the compilation.

```
cd /usr/lib
tar -zxvpf teTeX-lib-0.4pl7.tar.gz
tar -zxvf teTeX-src-0.4pl7.tar.gz
cd teTeX-src-0.4
```

Now edit the *Makefile*, and change one line.

```
BEFORE: TETEXDIR=/usr/local/teTeX
AFTER:  TETEXDIR=/usr/lib/teTeX
```

Then run `make world` and relax, it takes a good while. Afterwards, clean up the installation. Remove all the */usr/lib/teTeX/man/cat?/** man pages, compress all the */usr/lib/teTeX/info/** info pages, paste the header file */usr/lib/teTeX/info/dir* into the main info directory */usr/info/dir*, and symbolically link all the info pages in */usr/lib/teTeX/info* from */usr/info*. Edit the file */etc/profile* to add `/usr/lib/teTeX/bin/i486-linux` to the PATH, and to add `/usr/lib/teTeX/man` to the MANPATH. Get rid of any old binaries and/or TEX distribution you might have elsewhere. However, if there are some files that you want to keep that do not come with the new distribution, the copy them with their full path into */usr/local/lib/texmf*. For example, I keep the circle fonts there, and a *bibtex* style file I once modified.

Edit the file */usr/lib/teTeX/texmf.cnf* to add a few more directories to the search path for LATEX and TEX (for example, this allows LATEX to find the file *amssym.def*), and also enable a local TEX tree of files, where one puts files that don't come with the distribution, but that belong to TEX. The changed lines should read as below.

```
%TEXMFL         = /usr/local/tex.local
TEXMFL          = /usr/local/lib/texmf

% No local tree:
%TEXMFS         = !!$TEXMF
% Search local tree *after* the standard tree:
TEXMFS          = {!!$TEXMF,$TEXMFL}


TEXINPUTS.latex         = $KPSE_DOT:$TEXMFS/tex/{latex,latex209,generic,plain,
                                                  amstex}//
TEXINPUTS.latex209      = $KPSE_DOT:$TEXMFS/tex/{latex209,latex,generic,plain,
                                                  amstex}//

TEXINPUTS.tex           = $KPSE_DOT:$TEXMFS/tex{/plain,/generic,/amstex}//
```

Edit the script file */usr/lib/teTeX/bin/i486-linux/MakeTeXPK* and add the two mode lines below.

```
    204) MODE=GThreeFaxHigh;;
    360) MODE=bjtenex;;
```

Edit the script file */usr/lib/teTeX/texmf/xdvi/XDvi* to read below. I like 300 dpi as the default resolution.

```
;XDvi*mfmode: ljfour
;XDvi*pixelsPerInch: 600
;XDvi*shrinkFactor: 8
;XDvi*paper: a4
XDvi*mfmode: cx
XDvi*pixelsPerInch: 300
XDvi*shrinkFactor: 4
XDvi*paper: us
```

Edit the script file */usr/lib/teTeX/texmf/dvips/config/config.ps* and change the lines below. Commenting out the `"o |lpr"` has the effect that *dvips* now defaults to writing a ps-file, instead of writing to the printer. The other change of course sets the default resolution.

```
% How to print.
% o |lpr

% default resolution
% D 600
% M ljfour
D 300
M cx
```

Finally, write a short shell script to get the default font size for the Canon printer more conveniently, and call it *dvipslocal*. This file can contain more than one special mode, it should be called through symbolic links *dvipsbj* − > *dvipslocal* and *dvipslj* − > *dvipslocal*

/usr/bin/dvipslocal

```
#!/bin/sh
case ‘basename $0‘ in
   dvipsbj )
      echo "Dvips: using fonts for the Canon BJ series only (360dpi)."
      export DVIPSFONTS=/usr/lib/teTeX/texmf/fonts/pk/bjtenex//
      exec dvips -D360 $* ;;
   dvipslj )
      echo "Dvips: using fonts for the HP LaserJet series only (600dpi)."
      export DVIPSFONTS=/usr/lib/teTeX/texmf/fonts/pk/ljfour//
      exec dvips -D600 $* ;;
   # add any other special font driver you need often
   * )
      echo "Dvips: using default fonts (300dpi)."
      exec dvips $* ;;
esac
```

Missing fonts

Metafont usually creates fonts on the fly, but sometimes it doesn't work out. If the *mktexpk* script gets stuck and produces error messages, but also still produces a *.gf* file that is usable, then it is possible to run the whole thing by hand. Below is an example for the font bbm8 in toshiba mode.

```
mf '\mode:=toshiba; mag:=0+104/180; nonstopmode; input bbm8'
gftopk bbm8.104gf bbm8.104pk
install -m 644 bbm8.104pk /var/tmp/texfonts/pk/toshiba/public/bbm/
```

118 TEX-fonts

Files:
/usr/lib/texmf/fonts/Created_Fonts
/usr/lib/texmf/fonts/dc/misc/dcstdedt.tex

The information in here is no longer current.

Some Notes to the DC fonts, version 1.1, 1992-03-24

The DC fonts, which are the first (pre) release of the EC fonts, contain the text fonts of the *computer modern font family* with a coding scheme, that has been defined in the Cork meeting of TUG 1990 the *TEX 256 international character set*.

The fonts are generated without specific driver files, like *dcr10.mf*. Instead a interpolation algorithm is used to set up the needed parameters. A driver file has to contain only a specification of *gensize* and a call to the global driver file. A TEX run with the file *dcstdedt.tex* sets up a standard set of size specifying files which call the parameter files. By editing the file *dcstdedt.tex* the wished size set can be customized easily.

DCB	bold	DCSL	slanted
DCBX	bold extended	DCSLTT	slanted typewriter type
DCBXSL	bold extended slanted	DCSS	sans serife
DCBXTI	italic bold extended	DCSSBX	sans serife bold extended
DCCSC	caps and small caps	DCSSI	sans serife slanted
DCDUNH	dunhill	DCTCSC	typewriter caps and small caps
DCFF	funny roman	DCTI	italic
DCFI	funny italic	DCTT	typewriter type
DCFIB	fibonacci	DCU	italic unslanted
DCITT	italic typewriter type	DCVTT	variable typewriter type
DCR	roman		

Font compilation

All fonts are stored somewhere in */usr/lib/texmf/fonts*. They need to be compiled by running them through *metafont*. This is luckily all set up already so that each font directory has a shell script *create.sh* in it that calls the directories below, and at the lowest level finally *mf* and *gftopk* are called. So all it takes is one call from */usr/lib/texmf/fonts*.

However, life is not *that* nice and so some fonts don't work. They are all *dc* fonts. I did not install them. But if you do, here is what needs to be done. The routing information comes from the file */usr/lib/texmf/fonts/dc/misc/dcstedt.tex*, which if run through TEX, generates the file *mfbatch.bat*. The trick is now to change *dcstedt.tex* as to exclude the broken fonts. In the file *dcstedt.tex* I suggest you comment out the corresponding fonts where no sizes worked and delete the corresponding font sizes where only some don't work. Again, note that the file *mfbatch.bat* is created as a side effect of running TEX, no need to modify it.

Finally ran the thing from */usr/lib/texmf/fonts* with `sh < create.sh`. This input `"<"` symbol is important, otherwise the directories get goofed. There are still two fonts which come up with a need for user interaction, they bring up a `mf` window which needs to be clicked. Then I removed the files */usr/lib/texmf/fonts/dc/src/*.mf* which where created in the process. There are still quite a few steps to be done. I listed them below.

1. Copy the fonts from Utah into */usr/lib/texmf/fonts/utah/tfm* and *../pk*. They are backed up on floppy.
2. Set the permissions.

```
cd /usr/lib/texmf/fonts
chmod -R a+r,og-w *
find . -type d -print | xargs chmod 755
```

3. Clean out the local fonts that have been made so far. No need to keep them twice.

```
rm -f /var/texfonts/pk/localfont/*
```

4. Finally, make the fonts visible. Add path variables `TEXFONTS` and `XDVIFONTS` to */etc/profile*. Depending on your choice of fonts you will have a different path, of course.

```
export TEXFONTS=:/usr/lib/texmf/fonts/ams/tfm/:/usr/lib/texmf/fonts/cm/sauter
/tfm/:/usr/lib/texmf/fonts/cm/utilityfonts/committee/tfm/:/usr/lib/texmf/font
s/cm/utilityfonts/manualfonts/tfm/:/usr/lib/texmf/fonts/cm/latex/tfm/:/usr/li
b/texmf/fonts/symbol/logo/tfm/:/usr/lib/texmf/fonts/tfm/:/usr/lib/texmf/fonts
/utah/tfm/

export XDVIFONTS=:/usr/lib/texmf/fonts/ams/pk/:/usr/lib/texmf/fonts/cm/sauter
/pk/:/usr/lib/texmf/fonts/cm/utilityfonts/committee/pk/:/usr/lib/texmf/fonts/
cm/utilityfonts/manualfonts/pk/:/usr/lib/texmf/fonts/cm/latex/pk/:/usr/lib/te
xmf/fonts/symbol/logo/pk/:/usr/lib/texmf/fonts/pk/:/usr/lib/texmf/fonts/utah/
pk/:/usr/local/MuPAD/share/doc/hytex/fonts/
```

119 Texinfo

Files:
/usr/lib/texinfo/texinfo.tex
/usr/bin/: info makeinfo texindex texi2dvi install-info
/usr/man/man1/info.1.gz
/usr/info/: texinfo info.info.gz makeinfo.info.gz info-stnd.info.gz*
/usr/share/emacs/site-lisp/: info.el makeinfo.el texinfo.el texnfo-upd.el texnfo-tex.el texinfmt.el informat.el detexinfo.el new-useful-setqs

This is GNU software, obtainable from any GNU site. Unpack the sources, change the working directory, and run `./configure --prefix=/usr` and `make CFLAGS="-O2 -m486"`. Install with `make install`. Then compress man and info pages, update the directory for the info pages, and copy most of the *emacs/** files to the *site-lisp* directory.

120 Thesaurus

Files:
/usr/local/lib/thesaurus.gz
/usr/local/bin/thes

This is the Project Gutenberg Etext of Roget's Thesaurus No. Two. Project Gutenberg is based at Illinois Benedictine College. However, I had the hardest time finding the source text, and I finally located a copy at *http://www.vuw.ac.nz/non-local/gutenberg/texts/etext91/roget13a.txt*.

I compressed it and installed into */usr/local/lib/*. The script below simply uses a pager to display it. Use the thesaurus by using the search features of *less*.

thes

```
#!/bin/sh
PATH=${PATH}:/usr/local/bin
gzip -dc /usr/local/lib/thesaurus.gz | less
```

121 Toshiba Laptops

Files:
/etc/modules.conf
/dev/toshiba
/lib/modules/2.2.18/misc/toshiba.o
*/usr/local/toshiba/**
${HOME}/.tuxtimerc
/var/run/{ dispswitch.pid fan.pid hotkey.pid ownerstring.pid svpw.pid thotswap.pid tpasswd.pid tuxtime-conf.pid wmtuxtime.pid }

Sources obtainable at:
`http://www.buzzard.org.uk/toshiba`

These are the Toshiba Laptop utilities. For this to work, enable Toshiba support and loadable modules in the kernel configuration.

```
./configure --prefix=/usr/local/toshiba
make depend
make
su
mkdir -p /usr/local/toshiba/bin
mkdir -p  /usr/local/toshiba/man/man1 /usr/local/toshiba/man/man8
make install
```

Now strip the binaries, compress the manual pages, and link everything.

/etc/rc.d/rc.modules

The program does not seem to load the kernel module on demand, so I added it to the boot scripts. On the other hand, I don't know when ar what for that kernel module is at all needed, because the programs run fine without it.

```
# Toshiba Laptop support
if [ -f /lib/modules/${RELEASE}/misc/toshiba.o ]; then
  /sbin/modprobe toshiba
fi
```

/etc/modules.conf

The install script adds a line, and it is necessary to add another one by hand, see instructions below.

```
alias char-major-10-181 toshiba
options toshiba tosh_fn=0x62
```

Here is parts of the file *README.hotkey* that explain how to determine the correct BIOS address. The work "I" refers to the maintainer of the Toshiba Utilities package.
Finding the port takes a bit of experimentation. However I have tried to simplify it as much as possible with a shell script. In the src directory there is a shell script called fnfind. This needs to be run as root with
./fnfind
When run make sure that no keys are pressed, and then when prompted hold down Fn+F4 and keep it held down. This will only have the effect of changing the speaker volume. You have five seconds to hold down Fn+F4 after being prompted.
Hopefully the result from this should be something like this (which is from my laptop)
Make sure no keys are pressed
—
—
Hold down Fn+F4 (only changes speaker volume)
—
000060 9c 20 28 ff 14 ff 00 ff 3e ff ff ff 60 ff 00 ff
—
As you can see scanning along the line of number after holding down Fn+F4 we can see a value 3e. Count out from the start of the row of two character numbers starting a zero to 3e and add this to the first number. As a sanity check their should be no 3e when no keys are pressed.
In this case we get 0x68. Remember we are using hexadecimal so after 9 comes a. Currently two port locations are known for the Fn key.
*Make sure the toshiba kernel module is *unloaded* with "rmmod toshiba", then load the kernel module with port number as a parameter. On my laptop this would be*
insmod toshiba tosh_fn=0x68
Replace 0x68 with whatever the port number you have found from the first step is. Now try running the hotkey program. If it works then please email me the port number, and the output of "cat /proc/toshiba". I will then be able to modify the kernel module so that in future releases the module detects this automatically.
In the mean time you might wish to add the following line to /etc/conf.modules so that you don't have to specify the port number every time the module is loaded.
options toshiba tosh_fn=0x68
Again replace 0x68 with whatever the port number is for your notebook.

A patch file

The default behavior is to keep process id files in */var/tmp*, and I rather have them in */var/run*. Also, by default these files are dot-files, and hence not listed with a standard *ls* command. I removed those dots. Here is a patch that changes all that.

```
diff -rc toshutils-1.9.10/src/dispswitch.c toshutils-1.9.10~/src/dispswitch.c
*** toshutils-1.9.10/src/dispswitch.c Mon Feb  5 07:21:33 2001
--- toshutils-1.9.10~/src/dispswitch.c Sun Feb 25 19:38:40 2001
***************
*** 101,107 ****
  void CatchSignal(int x)
  {
   fprintf(stderr, "dispswitch: caught signal %d, cleaning up...\n", x);
!  unlink(_PATH_VARTMP PID_FILE);
   gtk_main_quit();
  }

--- 101,107 ----
  void CatchSignal(int x)
  {
   fprintf(stderr, "dispswitch: caught signal %d, cleaning up...\n", x);
!  unlink(_PATH_VARRUN PID_FILE);
   gtk_main_quit();
  }

***************
*** 160,166 ****
  #ifndef NO_X11
  void Deleted(GtkWidget *widget, gpointer *data)
  {
!  unlink(_PATH_VARTMP PID_FILE);
   gtk_main_quit();
  }

--- 160,166 ----
  #ifndef NO_X11
  void Deleted(GtkWidget *widget, gpointer *data)
  {
!  unlink(_PATH_VARRUN PID_FILE);
   gtk_main_quit();
  }

***************
*** 203,209 ****
  void Quit(int status)
  {
  #ifndef NO_X11
!  unlink(_PATH_VARTMP PID_FILE);
  #endif
   exit(status);
  }
--- 203,209 ----
  void Quit(int status)
  {
  #ifndef NO_X11
!  unlink(_PATH_VARRUN PID_FILE);
  #endif
```

```
   exit(status);
  }
***************
*** 296,303 ****
  #ifndef NO_X11
   /* check to see if a copy of Display Switch is already running */

!  if (!access(_PATH_VARTMP PID_FILE, R_OK)) {
!  if ((str = fopen(_PATH_VARTMP PID_FILE, "r" ))) {
   fscanf(str, "%d", &pid);
   fclose(str);

--- 296,303 ----
  #ifndef NO_X11
   /* check to see if a copy of Display Switch is already running */

!  if (!access(_PATH_VARRUN PID_FILE, R_OK)) {
!  if ((str = fopen(_PATH_VARRUN PID_FILE, "r" ))) {
   fscanf(str, "%d", &pid);
   fclose(str);

***************
*** 312,325 ****

   fprintf(stderr, "dispswitch: process %d appears to have"
   " died, continuing\n", pid);
!  unlink(_PATH_VARTMP PID_FILE);
   }
   }

   /* create the pid file */

   pid = getpid();
!  if ((str = fopen(_PATH_VARTMP PID_FILE, "w"))) {
   fprintf(str, "%d\n", pid);
   fclose(str);
   }
--- 312,325 ----

   fprintf(stderr, "dispswitch: process %d appears to have"
   " died, continuing\n", pid);
!  unlink(_PATH_VARRUN PID_FILE);
   }
   }

   /* create the pid file */

   pid = getpid();
!  if ((str = fopen(_PATH_VARRUN PID_FILE, "w"))) {
   fprintf(str, "%d\n", pid);
   fclose(str);
   }
```

```
diff -rc toshutils-1.9.10/src/fan.c toshutils-1.9.10~/src/fan.c
*** toshutils-1.9.10/src/fan.c Mon Feb  5 07:21:33 2001
--- toshutils-1.9.10~/src/fan.c Sun Feb 25 19:42:44 2001
***************
*** 91,97 ****

  enum { ON, OFF, STATUS, AUTO, TOGGLE };

! #define PID_FILE _PATH_VARTMP ".fan.pid"


  void FanOn(char *name)
--- 91,97 ----

  enum { ON, OFF, STATUS, AUTO, TOGGLE };

! #define PID_FILE _PATH_VARRUN "fan.pid"


  void FanOn(char *name)
diff -rc toshutils-1.9.10/src/hotkey.c toshutils-1.9.10~/src/hotkey.c
*** toshutils-1.9.10/src/hotkey.c Mon Feb  5 07:21:33 2001
--- toshutils-1.9.10~/src/hotkey.c Sun Feb 25 19:42:44 2001
***************
*** 114,120 ****
  #define FONT "-adobe-helvetica-bold-r-normal-*-12-*-*-*-*-*-*-*,\
  -*-*-*-r-normal--14-*-*-*-*-*"

! #define PID_FILE _PATH_VARTMP ".hotkey.pid"

  #define USAGE \
  "Usage: hotkey [OPTION]\n\n\
--- 114,120 ----
  #define FONT "-adobe-helvetica-bold-r-normal-*-12-*-*-*-*-*-*-*,\
  -*-*-*-r-normal--14-*-*-*-*-*"

! #define PID_FILE _PATH_VARRUN "hotkey.pid"

  #define USAGE \
  "Usage: hotkey [OPTION]\n\n\
diff -rc toshutils-1.9.10/src/misc.h toshutils-1.9.10~/src/misc.h
*** toshutils-1.9.10/src/misc.h Mon Feb  5 07:21:33 2001
--- toshutils-1.9.10~/src/misc.h Sun Feb 25 19:42:44 2001
***************
*** 29,35 ****
  #ifndef MISC_H
  #define MISC_H

! #define PID_FILE _PATH_VARTMP ".tuxtime-conf.pid"


  enum {
```

```
--- 29,35 ----
  #ifndef MISC_H
  #define MISC_H

! #define PID_FILE _PATH_VARRUN "tuxtime-conf.pid"


  enum {
diff -rc toshutils-1.9.10/src/ownerstring.c toshutils-1.9.10~/src/ownerstring.c
*** toshutils-1.9.10/src/ownerstring.c Mon Feb  5 07:23:28 2001
--- toshutils-1.9.10~/src/ownerstring.c Sun Feb 25 19:42:44 2001
***************
*** 35,41 ****
  #include"sci.h"
  #include"hci.h"

! #define PID_FILE _PATH_VARTMP ".ownerstring.pid"

  #define USAGE \
  "This utility can be used to set and change the owner string.\n\n\
--- 35,41 ----
  #include"sci.h"
  #include"hci.h"

! #define PID_FILE _PATH_VARRUN "ownerstring.pid"

  #define USAGE \
  "This utility can be used to set and change the owner string.\n\n\
diff -rc toshutils-1.9.10/src/svpw.c toshutils-1.9.10~/src/svpw.c
*** toshutils-1.9.10/src/svpw.c Mon Feb  5 07:21:33 2001
--- toshutils-1.9.10~/src/svpw.c Sun Feb 25 19:42:44 2001
***************
*** 55,61 ****
  struct termio saved;


! #define PID_FILE _PATH_VARTMP ".svpw.pid"

  #define USAGE \
  "This utility can be used to register the supervisor password and change the\n\
--- 55,61 ----
  struct termio saved;


! #define PID_FILE _PATH_VARRUN "svpw.pid"

  #define USAGE \
  "This utility can be used to register the supervisor password and change the\n\
diff -rc toshutils-1.9.10/src/thotswap.c toshutils-1.9.10~/src/thotswap.c
*** toshutils-1.9.10/src/thotswap.c Mon Feb  5 07:21:33 2001
--- toshutils-1.9.10~/src/thotswap.c Sun Feb 25 19:42:44 2001
***************
```

```
*** 75,81 ****
  #include"sci.h"
  #include"hci.h"

! #define PID_FILE _PATH_VARTMP ".thotswap.pid"

  static int id;
  static int locked = HCI_LOCKED;
--- 75,81 ----
  #include"sci.h"
  #include"hci.h"

! #define PID_FILE _PATH_VARRUN "thotswap.pid"

  static int id;
  static int locked = HCI_LOCKED;
diff -rc toshutils-1.9.10/src/tpasswd.c toshutils-1.9.10~/src/tpasswd.c
*** toshutils-1.9.10/src/tpasswd.c Mon Feb  5 07:21:33 2001
--- toshutils-1.9.10~/src/tpasswd.c Sun Feb 25 19:42:44 2001
***************
*** 52,58 ****
  struct termio saved;


! #define PID_FILE _PATH_VARTMP  ".tpasswd.pid"

  #define USAGE \
  "Register/remove the user password\n\n\
--- 52,58 ----
  struct termio saved;


! #define PID_FILE _PATH_VARRUN  "tpasswd.pid"

  #define USAGE \
  "Register/remove the user password\n\n\
diff -rc toshutils-1.9.10/src/wmtuxtime.h toshutils-1.9.10~/src/wmtuxtime.h
*** toshutils-1.9.10/src/wmtuxtime.h Mon Feb  5 07:21:33 2001
--- toshutils-1.9.10~/src/wmtuxtime.h Sun Feb 25 19:42:44 2001
***************
*** 28,34 ****


  #define DELAY 100000L
! #define PID_FILE _PATH_VARTMP ".wmtuxtime.pid"
  #define MINUTES(h,m) ((h*60)+m)

  enum {
--- 28,34 ----


  #define DELAY 100000L
```

```
! #define PID_FILE _PATH_VARRUN "wmtuxtime.pid"
  #define MINUTES(h,m) ((h*60)+m)

  enum {
```

122 Transfig

Files:
/usr/X11R6/bin/{fig2dev, fig2ps2tex, fig2ps2tex.sh, pic2tpic, transfig}
/usr/X11R6/man/man1/{fig2dev.1x.gz, fig2ps2tex.1x.gz, pic2tpic.1x.gz, transfig.1x.gz, fig2ps2tex.1x.gz, pic2tpic.1x.gz}
*/usr/doc/transfig/**

Sources obtainable at:
`ftp.x.org/contrib/applications/drawing_tools/transfig/`

Unpack the sources and make a few changes to the Imakefiles according to the local installations. After that, do the following.

```
cd transfig.3.1.2
xmkmf
cd fig2dev
xmkmf
cd dev
xmkmf
cd ../../tranfig
xmkmf
cd ..
make
make install
make install.man
gzip -9 /usr/X11R6/man/man1/*.1x
mkdir /usr/doc/transfig
cd doc
cp -R tpic.spec manual /usr/doc/transfig
```

The changes to the Imakefiles

```
diff transfig.3.2/fig2dev/Imakefile transfig.3.2.linux/fig2dev/Imakefile
57c57
< XCOMM #define USEXPM
---
> #define USEXPM
94c94
< XCOMM DDDVIPS = -DDVIPS
---
> DDDVIPS = -DDVIPS
103,104c103,104
< XCOMM DDNFSS = -DNFSS
```

```
< XCOMM DDEPSF = -DEPSF
---
> DDNFSS = -DNFSS
> DDEPSF = -DEPSF
109c109
< XCOMM DBMLIB = -ldbm
---
> DBMLIB = -ldbm
141c141
< JPEGINCDIR = /usr/include/X11
---
> JPEGINCDIR = /usr/local/include
diff transfig.3.2/fig2dev/dev/Imakefile transfig.3.2.linux/fig2dev/dev/Imakefile
24a25,26
> #define USE_JPEG
>
diff transfig.3.2/transfig/Imakefile transfig.3.2.linux/transfig/Imakefile
25c25
< XCOMM USELATEX2E = -DLATEX2E
---
> USELATEX2E = -DLATEX2E
```

123 Ttf2pt1

Files:
*/usr/local/ttf2pt1/**

Sources obtainable at:
`http://ttf2pt1.sourceforge.net`

This is a utility to transform True-Type-Fonts to Postscript Type 1 fonts. Edit the Makefile to set the compiler(-flags) and the location to install into.

124 Virtual Network Computer

Files:
*/usr/local/vnc/**
/usr/local/bin/{Xvnc, vncserver, vncpasswd, vncviewer}
*${HOME}/.vnc/**

Sources obtainable at:
`http://www.orl.co.uk/vnc/`
`http://www.uk.research.att.com/vnc/`

Get the linux binary distribution from the download page, install into */usr/local/vnc*, strip the binaries and link them from */usr/local/bin*. The permissions and ownerships are messed up, fix those. After running *vncserver* once, you might want to edit the *${HOME}/.vnc/xstartup* file. I prefer to simply replace it with a link to my *.xsession* file, or even the system wide */usr/X11/lib/X11/xinit/xinitrc* file.

125 Wavplay

Files:
/usr/local/bin/: wavplay wavrec xltwavplay wavplay_save
/usr/local/man/man1/wavplay.1.gz

Sources obtainable at:

```
http://home.ica.net/~wwg/
    wavplay-1.0.patch1.tar.gz
    wavplay-1.0.patch2.tar.gz
    wavplay-1.0.tar.gz
    wavplay.1
http://sunsite.unc.edu/pub/Linux/apps/sound/players
```

Version 1.0 now comes with two patches. Unpack and patch the sources as follows.

```
tar -zxvf wavplay-1.0.tar.gz
cd wavplay-1.0
tar -zxvf ../wavplay-1.0.patch1.tar.gz
./Applypatch1
tar -zxvf ../wavplay-1.0.patch2.tar.gz
./Applypatch2
```

Now edit the Makefile and add `-I/usr/X11/include` to the line `INCL = -I`. Then compile it (ignore the warnings) with `make`. Then strip and install the binaries into */usr/local/bin*. Finally write a short script called *wavplay_save* to enable Netscape to save wav-files before playing them.

```
#!/bin/sh
save $1 wav
wavplay $1
```

126 Wget

Files:
*/usr/local/wget/**

Sources obtainable at:

```
http://ftp.gnu.ai.mit.edu/pub/gnu/wget/
```

```
CFLAGS="-O2" ./configure --prefix=/usr/local/wget --disable-nls
make
make install
```

Strip the binary, compress the info pages, and link the whole thing from */usr/local*. Then make an entry into the info directory file */usr/info/dir*.

127 WordPerfect

Files:
*/usr/local/Corel/**
/usr/local/bin/xwp
/usr/local/man/man1/xwp.1.gz
*${HOME}/.wprc/**

Sources obtainable at:
`http://www.corel.com`

Get the linux binary distribution (which is called GUILG00) from the download page, and register. My registration key is `LW8XW-GnR3KGy1hK`. Also, you can get the manual from the same URL.

Corel warns that there is a possible security problem when installing the software as root. So I made a new user `corel` with group `corel` and home */usr/local/Corel*, who is never allowed to log in. This user will finally own all the files. However, I install under my usual user name, which is `mayer`. The instructions that come with the distribution say: untar the distribution into an empty directory, then untar all the archives you just created, and then run the script *Runme*. I also copied the manual into */usr/local/Corel*.

```
mkdir tmp
cd tmp
su
mkdir /usr/local/Corel
chown mayer.users /usr/local/Corel
exit
cp ../wp8gui.pdf /usr/local/Corel
tar -xvf ../GUILG00
for i in ?_*; do tar -xf $i; done
sh Runme
```

Answer all questions, install into */usr/local/Corel*, then select the printer *Hewlet Packard Laserjet 4 Postscript* and assign it to *lp*. Just run it once as the user `corel` to see if everything works, and it might finish up the installation this way, I don't remember. Also, write a short man page and wrapper script.

```
su
cd /usr/local/Corel
chown -R corel.corel .
chmod -R o-w .
su corel
cd wpbin
./xwp
```

The file *xwp.1*

```
.TH xwp 1 "6 February 1999"
.SH NAME
xwp \- WordPerfect word processor for X windows
.SH DOCUMENTATION
The manual is stored as
.B /usr/local/Corel/wp8gui.pdf
```

The file */usr/local/bin/xwp*

```
#!/bin/sh
cd /usr/local/Corel/wpbin
exec ./xwp -name xwp $*
```

The security warning by Corel

```
Corel Corporation has been made aware of a minor security issue that,
if left unmodified, could permit unwarranted access to files on the
computers of users of its Corel WordPerfect 8 for Linux software.

This issue, while minor, can be used to exploit a system in certain
circumstances.  The matter does not affect any users of Corel
WordPerfect 8 for Linux installed on non-networked or stand-alone
computers.

A small change in the installation process by network administrators
will nullify the unwarranted access issue, and that change is
documented below.

A fix can be enacted by users who have already installed Corel
WordPerfect 8 for Linux , but Corel recommends for optimal protection
that those in a networked environment delete the application and
reinstall the software using the method described below.

Corel has not been made aware of incidents where the security issue has
resulted in any unauthorized access of files at this time.

This issue only affects users of the no-charge download of Corel
WordPerfect 8 for Linux , purchasers of the stand-alone shrinkwrap
version of Corel WordPerfect 8 for Linux and the server edition of
Corel WordPerfect 8 for Linux /UNIX.

This is not an issue affecting users of Corel WordPerfect 8 for Linux on
any other platforms except Linux and UNIX.

Corel is working on a permanent solution to this problem, however in the
interim, the following work-around has been devised:

DO NOT INSTALL Corel WordPerfect 8 for Linux as "root".

Explanation of the problem:
```

When Corel WordPerfect 8 for Linux loads, it creates a directory called /tmp/wpc-<hostname> (where <hostname> is the host name of your computer) that has the UMASK 777.

Some necessary temporary files are stored in this folder, all containing the UMASK 666. When Corel WordPerfect 8 for Linux loads, it checks to see that these files exist, and if so, it will overwrite them.

However, Corel WordPerfect 8 for Linux doesn't check to see if these files have been replaced with sym-links of the same name. If this has been done (perhaps to compromise system security), Corel WordPerfect 8 for Linux will follow those sym-links and, if it has the rights to do so (which it will if you install the application as "root"), it will overwrite the file pointed to by the sym-link. This problem could be used to exploit a system.

Procedure to eliminate the problem:

If you have not installed Corel WordPerfect 8 for Linux yet, simply follow the instructions under "Installation Procedures".

If you have already installed Corel WordPerfect, we strongly recommend that you delete that installation and re-install the application. To delete a current installation of Corel WordPerfect 8 for Linux, execute the following instructions:

rm -r <path to Corel WordPerfect 8 for Linux Installation> (answer yes to the prompts)
rm -rf ~/.wprc
rm /etc/magic

Installation Procedures:

If you already have another user account set up on your system, then simply login under that user account, and run the installation program as you normally would.

NOTE: On most systems, you can only mount the CD-ROM drive as root, so you should do this before you login as a non-root user.

If you do not have a "non-root" user account setup on your system, the following will set that up (for RedHat Linux only):

1. Login as root

2. Type: adduser <username>
   where <username> can be any name you wish, but should be something short and easy to remember

3. Next type: passwd <username>
   where <username> is the same as what you chose for the previous

```
    step

  4. You will be prompted for a password for that user, and then asked to
     confirm that password

  5. If all went well, you will receive a message similar to the
     following:
        "passwd: all authentication tokens updated successfully"

To enact a fix to Corel WordPerfect 8 for Linux without re-installing the
software (this should only be attempted by advanced Linux users - Corel
recommends deleting the software and reinstalling using the previous
method).

This procedure will change the installation of Corel WordPerfect 8 for
Linux so that it can't be used to exploit the host system. This procedure
must be executed as root, and is recommended for advanced users only.

Command list:

cd <path to Corel WordPerfect 8 for Linux Installation>/..
chown <non-root user>: chown <non-root user>: rm -rf /tmp/wpc*

EXAMPLE:

cd /usr/local/wp8/..
chown johnd:johnd wp8 -R
chown johnd:johnd /etc/magic
rm -rf /tmp/wpc*

-------------------------------------------------------------------------
Copyright 1999 Corel Corporation. All rights reserved.
Last updated: Sun Jan 17 1999
```

128 WorkMan

Files:
/usr/openwin/bin/workman
/usr/openwin/lib/help/workman.info
/usr/openwin/man/man1/workman.1.gz
/usr/openwin/man/man5/workmandb.5.gz
*/usr/local/html/workman/**

Sources obtainable at:
`ftp://ftp.midwinter.com`

The location of the maintainer on the Internet has changed, but this is not yet reflected in the documentation. Adjust this with:

```
for i in DOCS/*; do
  sed -e 's/hyperion/midwinter/g' $i > /tmp/foo && cat /tmp/foo > $i
done
rm /tmp/foo
```

Edit *Makefile.linux*, so that you have the following:

```
#HACK = -DSBPCD_HACK
DBCFLAGS = -DLIBDB
LDLIBS = -lxview -lolgx -lX11 /usr/lib/libdb.so.2
```

Edit *plat_linux.c*, so that you have the following:

```
#ifdef SBPCD_HACK
#define DEFAULT_CD_DEVICE       "/dev/sbpcd"
#else
#define DEFAULT_CD_DEVICE       "/dev/cdrom"
#endif
```

Then compile and install.

```
make -f Makefile.linux
make -f Makefile.linux install
make -f Makefile.linux install.man
```

Now strip the binary and compress the man pages. Finally copy the HTML formatted documentation to */usr/local/html/workman*.

129 Xanim

Files:
*/usr/local/xanim/mods/**
/usr/X11/bin/xanim
/usr/X11/man/man1/xanim.1x.gz
/usr/local/bin/xanim_save

Sources obtainable at:
`http://xanim.va.pubnix.com/home.html`

```
            xanim2801.tgz.gz
            vid_cvid_2.0_linuxELFx86c5.tgz
            vid_cyuv_1.0_linuxELFx86c5.tgz
            vid_h261_1.0_linuxELFx86c5.tgz
            vid_h263_1.0_linuxELFx86c5.tgz
            vid_iv32_2.1_linuxELFx86c5.tgz
            vid_iv41_1.0_linuxELFx86c5.tgz
            vid_iv50_1.0_linuxELFx86c5.tgz
```

Install *libz* first, see Section 51.1.5. *Xanim* comes with precompiled dynamically loaded modules for specific video codecs. These need to be downloaded separately and installed into the default place, which is */usr/local/xanim/mods*.

Unpack the archive and cd into the newly created subdirectory. Read the documentation, in particular the file *docs/README.dll*.

First edit the *Imakefile*.

```
INCLUDES = -I$(INCDIR) $(XA_INCS) -I$(INCROOT)/X11
```

Then run the commands below.

```
xmkmf
make xanim
mkdir -p /usr/local/xanim/bin
install -s xanim /usr/local/xanim/bin
mkdir -p /usr/local/xanim/man/man1
install -m 644 docs/xanim.man /usr/local/xanim/man/man1/xanim.1
gzip -9 /usr/local/xanim/man/man1/xanim.1
```

I also wrote a little scirpt to use with Netscape to save downloaded movies before displaying them.

```
#!/bin/sh
save $1
xanim -CF4 $1
```

130 Xaudio

Files:
/usr/local/bin{*xaudio, mxaudio*}
/usr/local/man/man1/{*xaudio.1.gz, mxaudio.1.gz*}

Sources obtainable at:
`http://www.xaudio.com/download.html`

This is a binary distribution of an mpeg sound player. I install with:

```
install -m 644 -o root -g root README.mxaudio /usr/local/man/man1/mxaudio.1
install -m 644 -o root -g root README.xaudio /usr/local/man/man1/xaudio.1
gzip -9 /usr/local/man/man1/xaudio.1
gzip -9 /usr/local/man/man1/mxaudio.1
install -o root -g root xaudio /usr/local/bin
install -o root -g root mxaudio.dynamic /usr/local/bin/mxaudio
```

131 Xautolock and Xlock

Files:
/usr/X11R6/bin/xautolock

/usr/X11R6/bin/xlock
/usr/X11R6/man/man1/xautolock.1x.gz /usr/X11R6/man/man1/xlock.1x.gz

xautolock

Compiles right out of the box, `xmkmf`, `make`, and `make install`. Then gzip the man page. The program allows to automatically start any program after a given time of input-inactivity. The default is xlock, hence its name.

xlock, version 3.13

Also compiles more or less straight out of the box, however, it has by now 54 different screen savers. I took out all but eight of them, and left *dclock, galaxy, hyper, life, marquee, sphere, swarm,* and *blank*. Additionally there is *random*. For this, run `xmkmf` to create the Makefile, then edit this file and change several things. First, change the icon for *life*, so that it has the same as *life1d*. Then delete all the modes you don't want in the OBJ1 and SRC1 variables. Similarly, delete those modes from `mode.h` and `mode.c`, and delete the unnecessary information from the X resources file `XLock.ad` and from the man page `xload.man`. Link the bitmap for *life* with `make life.xbm` and compile with `make`. Install with `make install` and `make install.man`, then compress the man page, and delete any old installations. If desired, link the screen saver to a PopUp menu in the Fvwm window manager (`/usr/X11/lib/X11/fvwm/system.fvwmrc`). The entries are of the form

```
Exec    "Blank"         exec xlock -enablesaver -nolock -mode blank \&
```

Finally add an entry to the X startup file `/usr/X11/lib/X11/xinit/xinitrc`

```
(xautolock -locker "xlock -nice 20 -mode blank -enablesaver -lockdelay 5400" &)
```

This will automatically start the screen saver, and lock the screen after one hour, and the monitor shut-off feature of X is enabled.

132 Xbae Matrix Widget

Files:
*/usr/X11/lib/libXbae**
*/usr/X11/include/Xbae**
*/usr/X11/man/man3/Xbae**

Sources obtainable at:
`http://www.lesstif.org`
`http://www.xbae.syd.net.au`

Now distributed by the LessTif team, see Section 51.1.5.

```
CFLAGS="-O2" ./configure --prefix=/usr/X11
make
make install
gzip -9 /usr/X11/man/man3/*.3
ldconfig -v
makewhatis -v /usr/X11/man
```

I installed this for Xinvest only. Documentation comes with Lesstif (Section 51.1.5).

133 Xfig

Files:
/usr/X11R6/bin/xfig
/usr/X11R6/man/man1/xfig.1x.gz
/usr/X11R6/lib/X11/app-defaults/Fig
/usr/X11R6/lib/X11/app-defaults/Fig-color
*/usr/doc/xfig/**

Sources obtainable at:
ftp.x.org/contrib/applications/drawing_tools/xfig/

Unpack the sources and edit the Imakefile and edit the file Fig.ad.

```
make install
make install.man
gzip -9 /usr/X11/man/man1/xfig.1x
mkdir /usr/doc/xfig
cd Doc
cp FORMAT* MAKEPS TODO /usr/doc/xfig/
cd ..
tar -zcvf /usr/doc/xfig/Examples.tar.gz Examples/
```

The patch file

```
diff xfig.3.2/Imakefile xfig.3.2.linux/Imakefile
49c49
< XCOMM #define USEINSTALLEDJPEG
---
> #define USEINSTALLEDJPEG
69,70c69,70
< XCOMM #define USEXPM
< XCOMM #define USEXPM_ICON
---
> #define USEXPM
> #define USEXPM_ICON
82c82
< JPEGINCDIR = -I/usr/include/X11
---
> JPEGINCDIR = -I/usr/local/include
```

The changes of Fig.ad

Add the following two lines.

```
! reduce page height to fit a 1024x768 screen
Fig.pheight:                    7.75
```

134 Xinvest

Files:
/usr/X11R6/bin/Xinvest
/usr/lib/X11/app-defaults/Xinvest
*/usr/doc/Xinvest/**

This is a binary distribution of a financial program. Just untar the sources obtainable from the primary site `http://sunsite.auc.dk/xinvest/`, or from `ftp.x.org/contrib/applications`. Install the binary and Xresources file into the right place, and stuff the sample files into a directory in */usr/doc*. The Xresources file *Xinvest.ad* specifies a window for the reports that too small, I changed two lines to correct this.

```
*ReportScroll.width:                    650
*ReportScroll.height:                   300
```

135 Xforms

Files:
/usr/X11/lib/{libforms.so,libforms.so.89}
/usr/X11/include/forms.h
/usr/X11/man/man5/xforms.5.gz
/usr/X11/bin/colbrowser
/usr/X11/man/man1/colbrowser.1.gz

Sources obtainable at:
`http://world.std.com/ xforms/`
`ftp://ncmir.ucsd.edu/pub/xforms/linux-i386/elf/bxform-089-glibc2.1.tgz`
`ftp://sunsite.unc.edu/pub/Linux/X11/devel/xforms`

The binary distribution comes with a lot of files, only the header file and the library files are needed. I decided to keep one of the demos (a browser for the X windows RGB color library), and wrote a tiny man page for it also. As always when upgrading/installing dynamic libraries, don't forget to run `ldconfig` afterwards. It is necessary to install the library before compiling the demos. You might want to keep the old version (if you have it) because the versions are not entirely compatible and some older binary might need it. I made the whole thing into a slackware package.

136 Xldlas

Files:
/usr/local/bin/xldlas
/usr/local/man/man1/xldlas.1.gz
*/usr/local/xldlas/help/**

Sources obtainable at:
`http://metalab.unc.edu/pub/Linux/science/visualization/xldlas-0.85-source.tgz`

`http://www.a42.com/~thor/xldlas/`
`http://cascadia.ssc.com/~thor/xldlas/`

Xldlas (X, lies, damned lies, and statistics) is a basic statistical package that runs under the X Window System and is built around the XForms library. Details of use are available from within the program (see the Help menu). It needs the xforms libraries.

Installation is straight-forward. Unpack the shared-binary distribution, and put the binary *xldlas* into */usr/local/bin*, then compress the man page and put it into */usr/local/man/man1*, and finally move the *help* subdirectory into */usr/local/xldlas*.

If getting the source distribution, add the location of the *forms.h* header file to the compiler flags in the Makefile. Compile with `make`, install by hand as outlined above.

137 Xquote

Files:
/usr/X11/bin/Xquote
/usr/X11/lib/X11/app-defaults/Xquote

Sources obtainable at:
`http://sunsite.auc.dk/xinvest`
`xquote-1_1_tar.gz`

Compiles right out of the box:

```
xmkmf
make Makefile
make depend
make
make install
```

Notice: If you press the *About* button in the *Help* menu, then the program tries to connect to its home page on the WWW. If you are not online, then the program hangs a few minutes until the connection attempt times out.

Also, the Xresource file contains the specifications on how servers are providing the quotes. If this is not working out, then check the WWW page for an update.

138 Xscrabble

Files:
/usr/bin/X11/xscrabble
/usr/bin/X11/xscrab
/usr/lib/X11/app-defaults/Xscrabble
/usr/lib/X11/xscrabble

Get the sources at `http://www.belgarath.demon.co.uk/programs/` and install them somewhere. This source does not compile cleanly or even run on my Linux box. In particular it has a problem with the xpm background for the windows. So I decided to run the main window without any background, and the opening window has the standard window. To achieve all that apply a patch file that disables the calls to `load_data_pixmap`. The opening window however will work,

provided it is compiled with `-g` instead of `-O`. The patch file contains that change, too. Do the following.

```
tar -zxvf xscrabble.tar.gz
zcat /usr/src/xscrabble/xscrabble-patch | patch -p0
cd xscrabble
xmkmf
make Makefiles
make
strip src/xscrapple
su
install src/xscrapple /usr/bin/X11/
```

Now edit the `Imakefile` and change the compiler flag in the first line back to `-O`, and then compile again.

```
xmkmf
make clean
make Makefiles
make
strip src/xscrap
su
install src/xscrap /usr/bin/X11/
install -m 644 Xscrabble.ad /usr/lib/X11/app-defaults/Xscrabble
install -m 644 Xscrabble.ad /usr/lib/X11/app-defaults/Xscrabble
mkdir /usr/lib/X11/xscrabble
chmod 755 /usr/lib/X11/xscrabble
install -m 644 OSPD3.gz /usr/lib/X11/xscrabble/
install -m 644 scrabble_permutations /usr/lib/X11/xscrabble/
install -m 666 xscrabble.scores /usr/lib/X11/xscrabble/
```

The last file must be world writeable because it contains the high scores.

The patch file

```
diff -cr xscrabble/Imakefile xscrabble.ok/Imakefile
*** xscrabble/Imakefile Mon Mar 10 23:56:29 1997
--- xscrabble.ok/Imakefile Fri Sep 26 20:34:18 1997
***************
*** 1,5 ****

! #define PassCDebugFlags 'CDEBUGFLAGS= -O'
  #define IHaveSubdirs

  SUBDIRS = Xc src
--- 1,5 ----

! #define PassCDebugFlags 'CDEBUGFLAGS= -g'
  #define IHaveSubdirs

  SUBDIRS = Xc src
diff -cr xscrabble/src/main.c xscrabble.ok/src/main.c
*** xscrabble/src/main.c Mon Mar 10 00:41:24 1997
--- xscrabble.ok/src/main.c Fri Sep 26 20:34:02 1997
```

```
***************
*** 146,155 ****
   applicationShellWidgetClass,dpy[i],NULL);

   XtVaSetValues(topl[i],XtNtitle,player[i].name,NULL);
!  load_data_pixmap(slate_green_xpm,&bgxpm,topl[i]);
!
   set_board(topl[i],i);
!  set_icon_pixmap(xscrabble_xpm,topl[i]);
   acceptQuitPre(topl[i]);
   XtRealizeWidget(topl[i]);
   acceptQuitPost(topl[i]);
--- 146,156 ----
   applicationShellWidgetClass,dpy[i],NULL);

   XtVaSetValues(topl[i],XtNtitle,player[i].name,NULL);
!  /* load_data_pixmap(slate_green_xpm,&bgxpm,topl[i]);*/
!  /* FIXME pixmap is not loaded */
   set_board(topl[i],i);
!  /* set_icon_pixmap(xscrabble_xpm,topl[i]);*/
!  /* FIXME pixmap is not loaded */
   acceptQuitPre(topl[i]);
   XtRealizeWidget(topl[i]);
   acceptQuitPost(topl[i]);
```

139 Xv

Files:
/usr/X11/bin/{bggen,vdcomp,xcmap,xv,xvpictoppm}
/usr/X11/man/man1/{bggen.1.gz,vdcomp.1.gz,xcmap.1.gz,xv.1.gz,xvpictoppm.1.gz}

Sources obtainable at:
`http://www.trilon.com/xv/downloads.html`:

`xv-3.10a.tar.gz`
`mp-tiff-patch`
`xv-3.10a.JPEG-patch`
`xv-3.10a-png-1.2d.tar.gz`
`xvman310a-html.tar.gz` (optional documentation)

The support libraries.

These are *libjpeg*, *libz*, *libpng*, and *libtiff*. See Sections 51.1.5, 51.1.5, 51.1.5, and 51.1.5.

xv itself

First unpack the file *xv-3.10a.tar.gz* and cd into the newly created subdirectory. Remove the *jpeg* and *tiff* subdirectories if you want to use already installed versions of these support liibraries. Unpack the archive *xv-3.10a-png-1.2d.tar.gz*. Now apply the various patches.

```
patch -p0 </usr/src/xv/xv-3.10a.JPEG-patch
patch -p0 </usr/src/xv/mp-tiff-patch
patch -p0 </usr/src/xv/xv-3.10a.TIFF-patch
patch -p1 <xvpng.diff
```

Now edit the *Makefile* and change the lines listed below.

```
CC = gcc

CCOPTS= -O3 -fomit-frame-pointer

BINDIR = /usr/X11R6/bin
MANDIR = /usr/X11R6/man/man1
MANSUF = 1
LIBDIR = /usr/X11R6/lib

JPEG    = -DDOJPEG
# JPEGDIR = jpeg
# JPEGINC = -I$(JPEGDIR)
JPEGLIB = /usr/local/lib/libjpeg.so.6
# $(JPEGDIR)/jconfig.h:
#       cd $(JPEGDIR) ; ./configure CC='$(CC)'
# $(JPEGLIB):  $(JPEGDIR)/jconfig.h
#        cd $(JPEGDIR) ; make

PNGLIB = /usr/local/lib/libpng.so.3

TIFF    = -DDOTIFF
# TIFFDIR = tiff
# TIFFINC = -I$(TIFFDIR)
TIFFLIB = /usr/local/lib/libtiff.so.3
# $(TIFFLIB):
#       ( cd $(TIFFDIR) ; make CC='$(CC)' )

LIBS = -L/usr/X11/lib -lX11 $(JPEGLIB) $(TIFFLIB) $(PNGLIB) $(ZLIBLIB) -lm

MCHN = -DLINUX
```

Finally edit *xv.h* if you have a registered copy of xv. Also in this file uncomment the definition of `sys_errlist`, because this is now declared in a system header file. Then run `make` to compile the *xv* package. Install with `make install`. Then compress the man pages, and get rid of */usr/X11/lib/xvdocs.ps* because the html-formatted pages contain enough information anyways.

Finally install the documentation from */usr/doc*, then replace all the links with the compressed version of the links in the html pages, and compress all the html pages, like so:

```
cd /usr/doc
```

```
tar -zxf xvman310a-html.tar.gz
cd xvman310a
for i in *.html; do
  sed -e 's/html/html.gz/g' $i > foo; mv -f foo $i; gzip -9 $i
done
```

After that link the documentation from */usr/local/html/index.html*.

I now simply keep the PostScript version, and not the html formatted ones.

The INSTALL script

```
#!/bin/sh
# export XVSRC=/usr/src/graphics/xv
if [ "$XVSRC" = "" ] ; then
  echo "Please set the variable XVSRC to the location of your sources"
  echo "For a \"sh\"-type shell, use XVSRC=/your/path/ ; export XVSRC"
  echo "For a \"csh\"-type shell, use setenv XVSRC /your/path/"
  exit 0
fi
tar -zxvf $XVSRC/xv-3.10a.tar.gz
cd xv-3.10a/
gzip -dc $XVSRC/grabpatch.gz | patch -p0
gzip -dc $XVSRC/mp-tiff-patch.gz | patch -p0
gzip -dc $XVSRC/longname.patch.gz | patch -p1
gzip -dc $XVSRC/xpm.patch.gz | patch -p0
gzip -dc $XVSRC/deepcolor.patch.gz | patch -p0
gzip -dc $XVSRC/gifpatch.gz | patch -p0
gzip -dc $XVSRC/xv-3.10a.JPEG-patch.gz | patch -p0
gzip -dc $XVSRC/xv-3.10a.TIFF-patch.gz | patch -p0
tar -zxvf $XVSRC/xv-3.10a-png-1.2d.tar.gz
patch -p1 < xcmap.diff
# one hunk of the patch below fails
patch -p1 < xvpng.diff
# now fix that failing hunk
echo Fixing that failed hunk now, don\'t worry
gzip -dc $XVSRC/xvdir.patch.gz | patch -p0
gzip -dc $XVSRC/xvpng-fix2.patch.gz | patch -p0
# DON'T apply xvjpeg.diff, it's just a copy of the previous one
# Two hunks of the patch below fail
patch -p1 < xvtiff.diff
# now fix those two failing hunks
# with all that patching a new error is introduced, this patch fixes that too
echo Fixing those two failed hunks now, don\'t worry
gzip -dc $XVSRC/xvtiff.patch.gz | patch -p0
# I use the system wide installation of the jpeg and tiff libraries
rm -rf tiff jpeg
# Edit the Makefile, or apply the new patch
gzip -dc $XVSRC/Makefile.patch.gz | patch -p0
# Edit xv.h to show the registration string and to fix a duplicate definition
gzip -dc $XVSRC/xv.patch.gz | patch -p0
# Edit xvbrowse.c to show hidden files by default (uncomment next line here!)
# gzip -dc $XVSRC/xvbrowse.patch.gz | patch -p0
```

```
make
# make install
```

140 Xview

Files:
*/usr/openwin/**
*/usr/doc/xview/**
*/var/X11R6/lib/config/XV**

Sources obtainable at:
`ftp://sunsite.unc.edu/pub/Linux/libs/X/xview/`

This directory contains all the libraries and header files necessary to use xview applications. There are also a few applications which need to be installed separately, such as workman (see Section 102.1), for example. I downloaded the Slackware sources and compiled those, and then made a Slackware package, and install that with *pkgtool* (see Section 102.1). I needed to edit the install script and the *Makefile.linux* a bit. The install script in particular expects that there is an old version installed, from which it will save certain files. Get the whole subdirectory *xv* of the *source* directory of the slackware distribution, copy my patch and install script into it, and make the package from there with `sh xview-uwe.build`. Assemble the package with `cd /tmp; tar -zcvf xview.tgz usr var install`.

The patch for the Makefile

```
diff -cr xview-3.2p1.4/Makefile.Linux xview-3.2p1.4-uwe/Makefile.Linux
*** xview-3.2p1.4/Makefile.Linux Sun Nov 21 12:38:15 1999
--- xview-3.2p1.4-uwe/Makefile.Linux Sun Nov 21 11:44:30 1999
***************
*** 19,34 ****
   cd add_src/xcenter && make install
   cd add_src/meminfo && make clean
   cd add_src/xcenter && make clean
!  cp -a bitmaps images /usr/openwin/include
!  cp -a add_src/* /usr/openwin/share/src/xview/
!  cd add_files/usr/openwin/lib && cp -a oldmenus /usr/openwin/lib
!  cd /usr/openwin/lib && rm -f openwin-ss-ex openwin-uit openwin-xv-ex o
penwin-menu*
!  cd /usr/openwin/lib && ln -s oldmenus/* .
!  cd add_files/usr/openwin && cp -a etc /usr/openwin
!  cd add_files/usr/openwin/bin && cp -a svenv toolwait winsysck /usr/ope
nwin/bin
!  -rm -rf /usr/doc/packages/xview /usr/doc/packages/xv32base
!  mkdir -p /usr/doc/packages/xv32base
!  cd /usr/doc/packages/xv32base && ln -sf ../../../openwin/share/doc/xvi
ew
!  cd /usr/doc/packages && ln -sf xv32base xview
!  cd /usr/openwin/lib && ln -sf cmdtool shelltool
```

```
--- 19,34 ----
   cd add_src/xcenter && make install
   cd add_src/meminfo && make clean
   cd add_src/xcenter && make clean
!  cp -a bitmaps images /tmp/usr/openwin/include
!  cp -a add_src/* /tmp/usr/openwin/share/src/xview/
!  cd /usr/openwin/lib && cp -a oldmenus /tmp/usr/openwin/lib
!  cd /tmp/usr/openwin/lib && rm -f openwin-ss-ex openwin-uit openwin-xv-
ex openwin-menu*
!  cd /tmp/usr/openwin/lib && ln -s oldmenus/* .
!  -cd /usr/openwin && cp -a etc /tmp/usr/openwin
!  cd /usr/openwin/bin && cp -a svenv toolwait winsysck /tmp/usr/openwin/
bin
!  -rm -rf /tmp/usr/doc/packages/xview /tmp/usr/doc/packages/xv32base
!  mkdir -p /tmp/usr/doc/packages/xv32base
!  cd /tmp/usr/doc/packages/xv32base && ln -sf ../../../openwin/share/doc
/xview
!  cd /tmp/usr/doc/packages && ln -sf xv32base xview
!  cd /tmp/usr/openwin/lib && ln -sf cmdtool shelltool

diff -cr xview-3.2p1.4/Build-LinuxXView.bash xview-3.2p1.4-uwe/Build-LinuxXVie
w.bash
*** xview-3.2p1.4/Build-LinuxXView.bash Sun Nov 21 12:38:15 1999
--- xview-3.2p1.4-uwe/Build-LinuxXView.bash Sun Nov 21 11:12:46 1999
***************
*** 109,116 ****
        [ -e $OWDEST/lib/openwin-menu-std ] || mv $OWDEST/lib/openwin-menu $OW
DEST/lib/openwin-menu-std
        install -d $OWDEST/lib/xview $OWDEST/share/locale/C/props
        install -m 644 contrib/misc/openwin-menu* $OWDEST/lib
!       install -d /var/openwin/lib
!       install -m 755 contrib/misc/Xinitrc /var/openwin/lib
        (cd $OWDEST/lib && ln -sf ../../../var/openwin/lib/Xinitrc)
        install -m 755 contrib/misc/{openwin,owplaces} $OWDEST/bin
        install -m 644 contrib/misc/props-locale.C $OWDEST/share/locale/C/prop
s/C
--- 109,116 ----
        [ -e $OWDEST/lib/openwin-menu-std ] || mv $OWDEST/lib/openwin-menu $OW
DEST/lib/openwin-menu-std
        install -d $OWDEST/lib/xview $OWDEST/share/locale/C/props
        install -m 644 contrib/misc/openwin-menu* $OWDEST/lib
!       install -d $INSTPREFIX/var/openwin/lib
!       install -m 755 contrib/misc/Xinitrc $PREFIX/var/openwin/lib
        (cd $OWDEST/lib && ln -sf ../../../var/openwin/lib/Xinitrc)
        install -m 755 contrib/misc/{openwin,owplaces} $OWDEST/bin
        install -m 644 contrib/misc/props-locale.C $OWDEST/share/locale/C/prop
s/C
```

My install script xview-uwe.build

```
#!/bin/sh
```

```
# Build XView for Slackware
INSTPREFIX=/tmp
export INSTPREFIX
CWD=‘pwd‘
cd /tmp
tar xzvf $CWD/xview-3.2p1.4.src.tar.gz
cd xview-3.2p1.4
tar xzvf $CWD/clients.tar.gz
zcat $CWD/xview-3.2p1.4.diff.gz | patch -p1 -E
zcat $CWD/xview-3.2p1.4-uwe.diff.gz | patch -p1 -E
make -f Makefile.Linux compile
make -f Makefile.Linux install
installpkg $CWD/tmp_examples.tgz
cat $CWD/xinitrc.openwin > /tmp/var/X11R6/lib/xinit/xinitrc.openwin
chmod 755 /tmp/var/X11R6/lib/xinit/xinitrc.openwin
# mv /tmp/usr/X11R6/bin/xcenter /tmp/usr/openwin/bin/xcenter
mv /tmp/usr/openwin/bin/clock /tmp/usr/openwin/bin/xvclock
mv /tmp/usr/openwin/bin/msgfmt /tmp/usr/openwin/bin/xvmsgfmt
mv /tmp/usr/openwin/man/man1/clock.1x /tmp/usr/openwin/man/man1/xvclock.1x
mv /tmp/usr/openwin/man/man1/msgfmt.1 /tmp/usr/openwin/man/man1/xvmsgfmt.1
gzip -9 --force /tmp/usr/openwin/man/man1/*
mkdir -p /tmp/usr/doc/xview-3.2p1.4
cp -a LEGAL_NOTICE README README.LinuxELF README.X11R6 UNPACKING \
  /tmp/usr/doc/xview-3.2p1.4
chown root.root /tmp/usr/doc/xview-3.2p1.4/*
( cd /tmp/usr/doc/xview-3.2p1.4 ; ln -sf /tmp/usr/openwin/share/doc/xview . )
make -C /tmp/xview-3.2p1.4/add_src/xcenter
install -c -s /tmp/xview-3.2p1.4/add_src/xcenter/xcenter /tmp/usr/openwin/bin/
make -C /tmp/xview-3.2p1.4/add_src/meminfo
install -c -s /tmp/xview-3.2p1.4/add_src/meminfo/meminfo /tmp/usr/openwin/bin/
```

141 X11R6

Files:
/usr/X11R6/lib/X11/XF86Config
/usr/X11R6/lib/X11/xinit/xinitrc.fvwm
/var/X11R6/lib/fvwm/system.fvwmrc
~/.Xdefaults

When installing with `setup` (Section 102.1) choose the **XF86_S3** driver. *XF86Config* contains the monitor timings and video card information. There is a database file called *X386.modeDB* available from the Linux ftp sites. *xinitrc.fvwm* starts the programs which run when X windows comes up. This is also where the resource files are specified. *system.fvwmrc* contains the general X setup, like pull-down menus, background color, number of panels. *.Xdefaults* contains configuration data for window classes. I added stuff for `seyon` (Section 102.1).

XF86Config

Here are the necessary files and links.

```
lrwxrwxrwx   1 root  bin   /usr/X11/bin/X -> /var/X11R6/bin/X
lrwxrwxrwx   1 root  root  /var/X11R6/bin/X -> /usr/X11R6/bin/XF86_S3
-rwsr-xr-x   1 root  root  /usr/X11R6/bin/XF86_S3
lrwxrwxrwx   1 root  root  /usr/X11/lib/X11/xinit/xinitrc -> xinitrc.fvwm
```

This last file contains commands that are run when X comes up, like shells, clocks, other stuff. One might want to change some entries in /usr/X11/lib/X11/fvwm/system.fvwmrc to make the window manager look differently (in particular the IconPath and the PixmapPath were wrong in mine when I got it).

```
# This is /usr/X11/lib/X11/XF86Config
# modified by Uwe F. Mayer <mayer@math.brown.edu>
# $XConsortium: XF86Conf.cpp,v 1.5 95/01/16 13:16:10 kaleb Exp $
# $XFree86: xc/programs/Xserver/hw/xfree86/XF86Conf.cpp,v 3.11 1995/01/28
# 15:46:18 dawes Exp $
#
# Copyright (c) 1994 by The XFree86 Project, Inc.
#
# Permission is hereby granted, free of charge, to any person obtaining a
# copy of this software and associated documentation files (the "Software"),
# to deal in the Software without restriction, including without limitation
# the rights to use, copy, modify, merge, publish, distribute, sublicense,
# and/or sell copies of the Software, and to permit persons to whom the
# Software is furnished to do so, subject to the following conditions:
#
# The above copyright notice and this permission notice shall be included in
# all copies or substantial portions of the Software.
#
# THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
# IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
# FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.  IN NO EVENT SHALL
# THE XFREE86 PROJECT BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,
# WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF
# OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
# SOFTWARE.
#
# Except as contained in this notice, the name of the XFree86 Project shall
# not be used in advertising or otherwise to promote the sale, use or other
# dealings in this Software without prior written authorization from the
# XFree86 Project.
#

# **********************************************************************
# Refer to the XF86Config(4/5) man page for details about the format of
# this file. This man page is installed as
# /usr/X11R6/man/man5/XF86Config.5x
# **********************************************************************

# **********************************************************************
# Files section.  This allows default font and rgb paths to be set
# **********************************************************************
```

```
Section "Files"

# The location of the RGB database.  Note, this is the name of the
# file minus the extension (like ".txt" or ".db").  There is normally
# no need to change the default.

    RgbPath     "/usr/X11R6/lib/X11/rgb"

# Multiple FontPath entries are allowed (which are concatenated together),
# as well as specifying multiple comma-separated entries in one FontPath
# command (or a combination of both methods)

    FontPath    "/usr/X11R6/lib/X11/fonts/misc/"
    FontPath    "/usr/X11R6/lib/X11/fonts/Type1/"
    FontPath    "/usr/X11R6/lib/X11/fonts/Speedo/"
    FontPath    "/usr/X11R6/lib/X11/fonts/75dpi/"
    FontPath    "/usr/X11R6/lib/X11/fonts/100dpi/"

EndSection

# ****************************************************************
# Server flags section.
# ****************************************************************

Section "ServerFlags"

# Uncomment this to cause a core dump at the spot where a signal is
# received.  This may leave the console in an unusable state, but may
# provide a better stack trace in the core dump to aid in debugging

#    NoTrapSignals

# Uncomment this to disable the <Crtl><Alt><BS> server abort sequence
# This allows clients to receive this key event.

#    DontZap

# Uncomment this to disable the <Crtl><Alt><KP_+>/<KP_-> mode switching
# sequences.  This allows clients to receive these key events.

#    DontZoom

EndSection

# ****************************************************************
# Input devices
# ****************************************************************

# ****************************************************************
# Keyboard section
# ****************************************************************
```

```
Section "Keyboard"

    Protocol    "Standard"

# when using XQUEUE, comment out the above line, and uncomment the
# following line

#    Protocol   "Xqueue"

    AutoRepeat  500 5

# Let the server do the NumLock processing.  This should only be required
# when using pre-R6 clients
#    ServerNumLock

# Specifiy which keyboard LEDs can be user-controlled (eg, with xset(1))
#    Xleds      1 2 3

# To set the LeftAlt to Meta, RightAlt key to ModeShift,
# RightCtl key to Compose, and ScrollLock key to ModeLock:

#    LeftAlt     Meta
#    RightAlt    ModeShift
#    RightCtl    Compose
#    ScrollLock  ModeLock

EndSection


# **********************************************************************
# Pointer section
# **********************************************************************

Section "Pointer"

#   Protocol    "Microsoft"
    Protocol    "MouseSystems"
    Device      "/dev/mouse"

# When using XQUEUE, comment out the above two lines, and uncomment
# the following line.

#    Protocol   "Xqueue"

# Baudrate and SampleRate are only for some Logitech mice

#    BaudRate   9600
#    SampleRate 150

# Emulate3Buttons is an option for 2-button Microsoft mice

#    Emulate3Buttons
```

```
# ChordMiddle is an option for some 3-button Logitech mice

#    ChordMiddle

EndSection


# **********************************************************************
# Monitor section
# **********************************************************************

# Any number of monitor sections may be present

Section "Monitor"
  Identifier "MAG DX15Fe"
  VendorName "MAG"
  ModelName "DX15Fe"
  BandWidth 80 MHz
  HorizSync 30-64 KHz
  VertRefresh 50-100 Hz
# A generic VGA 640x480 mode (hsync = 31.5kHz, refresh = 60Hz)
#    Mode "640x480@60"
#         DotClock      25.175
#         HTimings      640 664 760 800
#         VTimings      480 491 493 525
#    EndMode
#
# Standard VESA at higher frequency, hsync about 40kHz
#
Modeline  "640x480@72"   31      640  664  704  832    480  489  492  520
#
# VESA 800x600@72Hz Non-Interlaced mode
# Horizontal Sync = 48kHz
# Timing: H=(1.12us, 2.40us, 1.28us) V=(0.77ms, 0.13ms, 0.48ms)
#
# name        clock   horizontal timing     vertical timing      flags
     Mode "800x600"
         DotClock     50
         HTimings     800  856  976 1040
         VTimings     600  637  643  666
         Flags        "+hsync" "+vsync"
     EndMode
#
# VESA 1024x768@70Hz Non-Interlaced mode
# Horizontal Sync=56.5kHz
# Timing: H=(0.32us, 1.81us, 1.92us) V=(0.05ms, 0.14ms, 0.51ms)
#
# name        clock   horizontal timing     vertical timing      flags
    Mode "1024x768"
        DotClock      75
        HTimings      1024 1048 1184 1328
```

```
        VTimings       768  771  777  806
        Flags          "-hsync" "-vsync"
    EndMode
#
# Modeline "1152x900" 80    1152 1184 1228 1448    900  901  903  922
EndSection


# **********************************************************************
# Graphics device section
# **********************************************************************

# Any number of graphics device sections may be present

# copied from README.S3
#     S3 864, S3 86C716 SDAC RAMDAC and Clockchip
#        Diamond Stealth 64 DRAM
#        8, 15/16 and 24 bpp
#        ClockChip "s3_sdac"  (should be detected automatically)
Section "Device"
        Identifier  "Diamond Stealth 64 DRAM"
        VendorName  "Diamond Stealth"
        ClockChip   "s3_sdac"
        VideoRam    2048
        Chipset     "mmio_928"
        Option      "power_saver"
EndSection


# **********************************************************************
# Screen sections
# **********************************************************************

# The accelerated server S3

Section "Screen"
     Driver             "accel"
     Device             "Diamond Stealth 64 DRAM"
     Monitor            "MAG DX15Fe"
     Subsection         "Display"
          Depth         8
          Modes         "800x600" "1024x768" "640x480@72"
          ViewPort      0 0
          Virtual       1280 1024
     EndSubsection
     Subsection         "Display"
          Depth         16
          Modes         "800x600" "1024x768"
          ViewPort      0 0
          Virtual       1024 768
     EndSubsection
     Subsection         "Display"
```

```
          Depth         24
          Modes         "800x600"
          ViewPort      0 0
          Virtual       800 600
     EndSubsection
EndSection
```